

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютерних та інформаційно-пошукових систем»**

спеціальності 121 Інженерія програмного забезпечення

на тему: «Веб-сервіс для аналізу коментарів на YouTube»

Виконав:

студент IV курсу, групи КП-61

Брусенцов Юрій Олексійович _____

Керівник:

Старший викладач кафедри ПЗКС, к.т.н.,

Люшенко Леся Анатоліївна _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович _____

Рецензент:

Доцент кафедри ММСА ІПСА, к.ф.-м.н., доцент,

Шубенкова Ірина Анатоліївна _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення комп'ютерних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2019 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Брусенцову Юрію Олексійовичу

1. Тема проєкту «Веб-сервіс для аналізу коментарів на YouTube», керівник проєкту Люшенко Леся Анатолівна, ст. викладач, к.т.н., затверджені наказом по університету від «25» травня 2020 р. №1181-С.
2. Термін подання студентом проєкту «___» червня 2020 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - аналіз існуючих рішень поставленої задачі;
 - обґрунтування вибору засобів реалізації;
 - розроблення веб-сервісу для аналізу коментарів на YouTube;
 - аналіз розробленого веб-сервісу для аналізу коментарів на YouTube.
5. Перелік обов'язкового графічного матеріалу:
 - діаграма компонентів;
 - схема бази даних;
 - розроблений інтерфейс користувача (плакат);
 - результат роботи модулю для побудови графіків (плакат).

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	14.11.2019	
2.	Розроблення та узгодження технічного завдання	28.11.2019	
3.	Аналіз аналогів системи та розробка вимог	21.12.2019	
4.	Підготовка матеріалів першого розділу дипломного проєкту	30.12.2019	
5.	Проектування архітектури системи	15.01.2020	
6.	Аналіз та вибір засобів реалізації	24.01.2020	
7.	Створення структури бази даних	30.01.2020	
8.	Розробка системи	15.02.2020	
9.	Тестування системи, виправлення недоліків	01.04.2020	
10.	Підготовка матеріалів другого розділу дипломного проєкту	14.04.2020	
11.	Підготовка матеріалів третього розділу дипломного проєкту	21.04.2020	
12.	Підготовка матеріалів четвертого розділу дипломного проєкту	28.04.2020	
13.	Оформлення документації дипломного проєкту	07.05.2020	

Студент

Юрій БРУСЕНЦОВ

Керівник проєкту

Леся ЛЮШЕНКО

АНОТАЦІЯ

Даний дипломний проєкт присвячено розробці програмного застосунку призначеного для спрощення та автоматизації аналізу, пошуку, фільтрації, візуалізації статистики та лінгвістичного аналізу зібраних коментарів із відео на YouTube заданого користувачем.

У даному дипломному проєкті було розроблено: архітектуру клієнтської та серверної частини веб-сервісу, системи керування базами даних, модуль для збирання коментарів і лінгвістичного аналізу їх повідомлень та налаштовано роботу із сторонніми сервісами для забезпечення авторизації та збору даних про коментарі та відео.

Розроблювана система представлена у вигляді веб-сервісу, а саме веб-сайту із адаптивним графічним інтерфейсом користувача. Сервіс дозволяє користувачу переглядати усі коментарі із YouTube-відео, виконувати пошук, фільтрацію, надає проаналізовані дані про текст коментарю, а саме тональність тексту та мову на якій він написаний. Статистична інформація про аналізовані коментарі надається сервісом у вигляді графіків та діаграм.

ABSTRACT

This diploma project deals with the development of a software application designed to simplify and automate the analysis, search, filtering, visualization of statistics and linguistic analysis of collected comments from videos on YouTube given by the user.

In this project were developed: the architecture of the client and server part of the web service, database management system, module for collecting comments and linguistic analysis of comments messages and set up work with third-party services to authorize and collect data about comments and videos.

The developed system is presented in the form of a website with an adaptive graphical user interface. The service allows the user to view all comments from the YouTube video, search and filter, provides analyzed data about the text of the comment, namely the emotion tone of the text and the language in which it is written. Statistical information about the analyzed comments is provided by the service in the form of graphs and charts.

АННОТАЦИЯ

Данный дипломный проект посвящен разработке программного обеспечения что предназначен для упрощения и автоматизации анализа, поиска, фильтрации, визуализации статистики и лингвистического анализа собранных комментариев с заданного пользователем видео на YouTube.

В данном дипломном проекте было разработано: архитектуру клиентской и серверной частей веб-сервиса, системы управления базами данных, модуль для сбора комментариев и лингвистического анализа их сообщений и настроено работу с внешними сервисами для обеспечения авторизации и сбора данных про комментарии и видео.

Разрабатываемая система представлена в виде веб-сервиса, а именно веб-сайта с адаптивным графическим интерфейсом пользователя. Сервис позволяет пользователю просматривать все комментарии из YouTube-видео, выполнять поиск, фильтрацию, предоставляет проанализированы данные о тексте комментария, а именно тональность текста и язык, на котором он написан. Статистическая информация об анализированных комментариях предоставляется сервисом в виде графиков и диаграмм.

ДП.045440-01-90 Веб-сервіс для аналізу коментарів на YouTube. Відомість проєкту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проєкту		
ДП.045440-02-91	Веб-сервіс для аналізу	5	
	коментарів на YouTube.		
	Технічне завдання		
ДП.045440-03-81	Веб-сервіс для аналізу	53	
	коментарів на YouTube.		
	Пояснювальна записка		
ДП.045440-04-51	Веб-сервіс для аналізу	4	
	коментарів на YouTube.		
	Програма та методика		
	тестування		
ДП.045440-05-34	Веб-сервіс для аналізу	14	
	коментарів на YouTube.		
	Керівництво користувача		
ДП.045440-06-99	Веб-сервіс для аналізу	1	
	коментарів на YouTube.		
	Архітектура системи.		
	Діаграма компонентів		
	веб-сервісу. ER-діаграма		
ДП.045440-07-99	Веб-сервіс для аналізу	1	
	коментарів на YouTube.		
	Архітектура системи.		
	Схема бази даних		
	веб-сервісу. ER-діаграма		

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ Іван ДИЧКА

“ ____ ” _____ 2019 р.

ВЕБ-СЕРВІС ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

Технічне завдання

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Юрій БРУСЕНЦОВ

ЗМІСТ

1. Найменування та галузь застосування	3
2. Підстава для розроблення	3
3. Призначення розробки.....	3
4. Вимоги до програмного продукту	3
5. Вимоги до проєктної документації.....	4
6. Етапи проєктування	5
7. Порядок тестування розробки.....	5

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Веб-сервіс для аналізу коментарів на YouTube.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розроблюване програмне забезпечення призначене для спрощення та автоматизації аналізу, пошуку, фільтрації, візуалізації статистики та лінгвістичного аналізу зібраних коментарів із відео на YouTube заданого користувачем.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Система має відповідати наступним функціональним вимогам:

- можливість надавати відео для аналізу за допомогою посилання на YouTube;
- можливість переглянути коментарі у вигляді списку та їх загальну кількість;
- можливість пошуку коментарів за входженням текстового рядка;
- можливість сортування коментарів за кількістю лайків та датою створення коментарю;

- можливість відображення тональності та мови написання тексту коментарю;
- забезпечення візуалізації статистики обраного відео у вигляді графіків та діаграм.

Вимоги до безпеки програми:

- відсутність можливості JavaScript-ін'єкцій у додатку;
- авторизація користувача за допомогою Google OAuth 2.0 API;
- захист від ін'єкцій до бази даних.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- пояснювальна записка;
- програма та методика тестування;
- керівництво користувача;
- креслення:
 - «Діаграма компонентів системи»;
 - «Результат роботи модулю для побудови графіків».

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою проєкту.....	14.11.2019
Розроблення та узгодження технічного завдання.....	28.10.2019
Аналіз аналогів системи та розробка вимог.....	21.12.2019
Підготовка матеріалів першого розділу.....	30.12.2019
Проектування архітектури системи.....	15.01.2020
Аналіз та вибір засобів реалізації.....	24.01.2020
Створення структури бази даних.....	30.01.2020
Розробка системи.....	15.02.2020
Тестування системи, виправлення недоліків.....	01.04.2020
Підготовка матеріалів другого розділу.....	14.04.2020
Підготовка матеріалів третього розділу.....	21.04.2020
Підготовка матеріалів четвертого розділу.....	28.04.2020
Оформлення документації дипломного проєкту.....	07.05.2020

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«__» _____ 2020 р.

ВЕБ-СЕРВІС ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

Пояснювальна записка

ДП.045440-03-81

«ПОГОДЖЕНО»

Керівник проекту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Юрій БРУСЕНЦОВ

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП.....	7
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	9
1.1. Загальні положення та аналіз предметної області.....	9
1.2. Аналіз аналогічних застосунків.....	10
1.3. Актуальність розробки веб-сервісу для аналізу коментарів на YouTube	14
1.4. Загальні вимоги до системи.....	15
1.5. Висновки до розділу	16
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ	18
2.1. Обґрунтування вибору фреймворку для програмування серверної частини.....	18
2.2. Обґрунтування вибору фреймворку для програмування клієнтської частини.....	20
2.3. Обґрунтування вибору системи керування базами даних	22
2.4. Висновки до розділу	24
3. РОЗРОБЛЕННЯ ВЕБ-СЕРВІСУ ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE.....	26
3.1. Загальний опис системи	26
3.2. Аналіз функціональних вимог до програмного додатка.....	28
3.3. Архітектура системи.....	29
3.4. Особливості реалізації.....	35
3.5. Висновки до розділу	40
4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБ-СЕРВІСУ ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE	42
4.1. Аналіз розробленого веб-додатку	42
4.2. Тестування розробленого веб-додатку	43
4.3. Порівняння розробки із існуючими аналогами	44
4.4. Рекомендації щодо подальшого вдосконалення.....	45
4.5. Висновки до розділу	46
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	50
ДОДАТКИ.....	53

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

YouTube – відеохостинг, що надає користувачам послуги розміщення та перегляду відеоматеріалів.

Блог – це веб-сайт, головний зміст якого – записи, зображення чи мультимедіа, що регулярно додаються.

Блогер – людина, що є автором блогу (у даному проєкті – автором YouTube-каналу).

Аналіз тональності тексту (англ. Sentiment analysis) – клас методів контент-аналізу в комп'ютерній лінгвістиці, призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті.

Лайк – базове поняття в соціальних сервісах, що поширилося разом з соціальними мережами. Умовний вираз схвалення матеріалу, виражається натисненням однієї кнопки

Telegram – багатоплатформовий месенджер, що дозволяє обмінюватися повідомленнями і медіафайлів багатьох форматів.

Excel – програма для роботи з електронними таблицями, створена корпорацією Microsoft для Microsoft Windows.

Веб-сервер – сервер, який приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, та видає їм HTTP-відповіді.

HTTP (англ. HyperText Transfer Protocol) – протокол прикладного рівня передачі даних у вигляді гіпертекстових документів в форматі HTML.

HTML (англ. HyperText Markup Language) – стандартизована мова розмітки гіпертекстових документів для мережі Інтернет.

Веб-клієнт – програмний компонент, який посилає запити веб-серверу.

API (англ. Application Programming Interface) – опис інструментів для взаємодії із програмою.

Фреймворк (англ. framework) – програмна платформа, що визначає структуру програмної системи.

JavaScript – мультипарадигмальна мова програмування. Є реалізацією стандарту ECMAScript.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду.

MVC (англ. Model-View-Controller) – схема поділу даних програми, користувацького інтерфейсу і керуючої логіки на три окремих компоненти: модель, представлення і контролер.

Windows – сімейство комерційних операційних систем корпорації Microsoft.

JSX (англ. JavaScript XML) – це розширення синтаксису JavaScript, яке дозволяє використовувати схожий на HTML синтаксис для опису структури інтерфейсу.

DOM (англ. Document Object Model) – програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML, XHTML і XML-документів, а також зміну вмісту, структури і оформлення таких документів.

TypeScript – мова програмування що розширює можливості JavaScript та додає можливість явного статичного призначення типів.

СКБД – система керування базами даних.

Резидентна СКБД – база даних, що розміщується в оперативній пам'яті.

Кеш – проміжний буфер даних з швидким доступом до нього, який містить інформацію, що може бути запрошена з найбільшою ймовірністю.

Кешування – занесення даних у кеш.

Linux – сімейство Unix-подібних операційних систем на базі ядра Linux, що включають той чи інший набір утиліт і програм проєкту GNU, та інші компоненти.

MacOS – операційна система виробництва Apple.

SQL (англ. Structured Query Language) – мова управління базами даних для реляційних баз даних.

NoSQL (англ. non SQL) – база даних, яка забезпечує механізм зберігання та видобування даних відмінний від підходу реляційних баз даних.

JSON (англ. JavaScript Object Notation) – текстовий формат обміну даними, заснований на JavaScript.

BSON (англ. Binary JavaScript Object Notation) – бінарна форма подання JSON.

Веб-сокет (англ. WebSocket) – протокол зв'язку поверх TCP-з'єднання, призначений для обміну повідомленнями між браузером і веб-сервером в режимі реального часу.

SPA (англ. Single Page Application) – веб-додаток або веб-сайт, який використовує єдиний HTML-документ як оболонку для всіх веб-сторінок.

SSR (англ. Server Side Rendering) – рендеринг клієнтської частини веб-додатку на сервері;

CSR (англ. Client Side Rendering) – рендеринг додатку на стороні клієнта;

MVC (англ. Model-View-Controller) - шаблон проєктування архітектури додатку. Шаблон розділяє логіку на три окремих компоненти:

модель, уявлення і контролер - таким чином, що модифікація кожного компонента може здійснюватися незалежно.

MVP (англ. Model-View-Presenter) – шаблон проєктування, похідний від MVC, який використовується в основному для побудови призначеного для користувача інтерфейсу.

MVVM (англ. Model-View-ViewModel) – шаблон проєктування архітектури додатку. Шаблон полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки.

Socket.IO – JavaScript-бібліотека для обміну даними в реальному часі. Складається з двох частин: клієнтської, яка запускається в браузері і серверної для Node.js.

ODM (англ. Object-Document Mapping) – технологія програмування, яка зв'язує бази даних з концепціями мови програмування, створюючи «віртуальну документоорієнтовану базу даних».

PDF (англ. Portable Document Format) – міжплатформенний відкритий формат електронних документів, розроблений компанією Adobe Systems.

PaaS (англ. Platform as a Service) – модель надання хмарних обчислень, при якій споживач отримує доступ до використання інформаційно-технологічних платформ.

ВСТУП

На сьогоднішній день, 59% всього населення планети має доступ до мережі Інтернет [1]. Дослідження Google Connected Consumer Study 2017 показало, що 66% українців регулярно користуються мережею Інтернет. Найпопулярнішими заняттями онлайн є: використання пошукових систем (91%), соціальних мереж (83%) і перегляд відео онлайн (84%) [2]. YouTube є найпопулярнішим відео-хостингом та другим по відвідуваності веб-сайтом у світі, яким щомісячно користуються 2 мільярди користувачів [3]. Популярність даного відеохостингу спричинена тим, що сайт є доступним та на ньому можна знайти відео на будь-яку тему.

Висока кількість користувачів відео-хостингу YouTube мотивує безліч людей з усіх куточків світу створювати свої відео-блоги. Такі люди називаються блогерами. Ведення відео-блогу, на сьогоднішній день, це не лише хобі чи розвага, для значної частки блогерів це повноцінна робота, адже YouTube має велику кількість користувачів, що робить не тільки платформу привабливою для рекламодавців, а і самих блогерів. Для розуміння та підтримки інтересу аудиторії і, як наслідок, рекламодавців, блогери користуються інструментами для збору статистики. На ринку програмних застосунків, які орієнтуються на аналіз статистики, є велика кількість програмних рішень, проте досі не існує програмного забезпечення, що аналізує коментарі за змістом їх повідомлень.

Проект, що розробляється у рамках дипломної роботи, присвячено створенню програмного забезпечення яке орієнтоване на аналіз коментарів на відеохостингу YouTube, що буде реалізовано у вигляді веб-сервісу. Даний сервіс буде надавати можливості лінгвістичного аналізу повідомлень, сортування, фільтрації та іншої роботи із коментарями. Саме таких функціональних можливостей не вистачає популярним сервісам орієнтованим на аналіз YouTube-каналів для забезпечення повноцінного охоплення статистичних даних.

Метою даного проєкту є створення веб-додатку, призначеного для спрощення та автоматизації аналізу, пошуку, фільтрації, візуалізації статистики та лінгвістичного аналізу зібраних коментарів із відео на YouTube заданого користувачем.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Загальні положення та аналіз предметної області

На сьогоднішній день YouTube є найпопулярнішим відеохостингом. Формат надання інформації у формі відео легше сприймається споживачем, аніж письмові статті або аудіо подкасти. Відеохостинги мають інструмент для зворотного зв'язку із своєю аудиторією у вигляді коментарів. Кожен користувач може висловити свою думку щодо відео, пропозицію або побажання до автора відеоролика. У популярних каналів велика кількість коментарів, тому з кожним днем все більше блогерів стикаються із проблемою аналізу своєї аудиторії та її реакції на контент.

Кількість коментарів – це важливий критерій, від якого залежить просування відео внутрішніми алгоритмами серед користувачів сервісу, що спричинює те, що блогери різними методами спонукають своїх глядачів до написання коментарів. Кожен з цих методів має різну силу впливу на глядачів, яку необхідно досліджувати, оскільки аудиторія кожного блогера індивідуальна.

Відсутність аналізу коментарів може стати причиною втрати зв'язку між блогером та його глядачами. Це може призвести до зниження активності на каналі та втрати аудиторії, що, в свою чергу, знизить привабливість блогера для рекламодавців.

Підсумовуючи викладене вище, зазначемо, що коментарі є важливою складовою для життя каналу, тож, вони потребують детального аналізу.

Основними проблемами при аналізуванні коментарів, з якими зіштовхуються блогери є:

- необхідність прочитати всі коментарі. Оскільки кількість коментарів у популярних відео може перевищувати десятки тисяч, то перечитати їх усі є значною проблемою;
- відсутність інструменту для аналізу коментарів у конкурентів. За допомогою аналізу коментарів у інших блогерів можна виявити

певні тенденції розвитку каналу, що аналізувався, та загальний настрій аудиторії у коментарях;

- відсутність пошуку по коментарям. Щоб знайти коментар із інформацією, що цікавить користувача, він має проглядати тисячі коментарів або використовувати сторонні додатки;
- відсутність фільтрації коментарів;
- відсутність візуалізації статистичних даних про коментарі. Візуалізовані дані легко сприймаються, за їх допомогою можна швидко передати велику кількість інформації.

1.2. Аналіз аналогічних застосунків

Проаналізуємо найпопулярніші застосунки наявні на ринку, що мають функціональні можливості аналізу YouTube-каналів. Для аналізу було обрано застосунки «Popsters», «SeeZisLab» та «LiveDune».

1.2.1. Додаток Popsters

«Popsters» – сервіс аналітики спільнот у соціальних мережах. Сервіс реалізований у вигляді веб-сайту, та у вигляді розширення до браузеру Google Chrome. У користувача сервісу є можливість проаналізувати як свій канал, так і канали конкурентів параметрами, за такими як: лайки, репости, перегляди, кількість публікацій, хештеги, тощо. Також сервіс має функціональні можливості для роботи із коментарями, у додатку є функціональні можливості для побудови графіків, що відображують сумарну кількість всіх коментарів у аналізованому періоді. Графік можна масштабувати, а також включати фільтрацію за днем публікації. Дані функціональні можливості стануть у нагоді користувачу, якщо він захоче оцінити загальну статистику кількості коментарів у обраному періоді.

Сервіс є платним та коштує приблизно \$55 за рік користування для аналізу однієї соціальної мережі та одного користувача. Надання аналізу нового користувача та додаткової соціальної мережі потребує доплати [4]. Інтерфейс веб-сайту наведено на рис. 1.

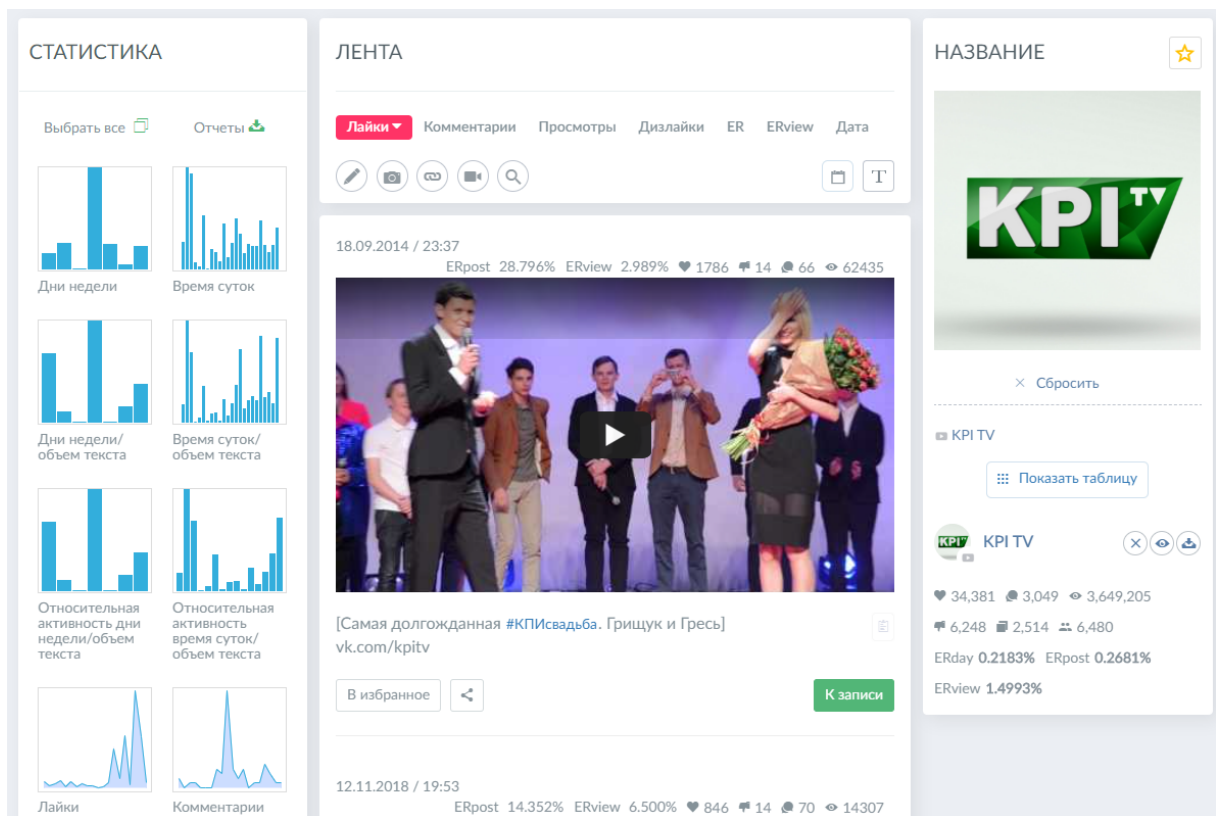


Рис. 1. Сторінка аналізу каналу на сервісі «Popsters»

Цей додаток буде корисний користувачам, які хочуть аналізувати багато сторінок у соціальних мережах одночасно. Проте, майже всі функціональні можливості, розроблені у «Popsters», має YouTube у своєму розділі «YouTube аналітика». Обробка коментарів йде лише у вигляді статистики, у аналізі не використовується сам текст коментарів.

1.2.2. Додаток SeeZisLab

«SeeZisLab» – сайт, який позиціонує себе як платформа для аналітики і статистики YouTube каналів. Додаток дозволяє збирати статистику, можливість заробітку та основних характеристик будь-якого каналу на YouTube. Сервіс має можливість показувати теги, соціальні характеристики та рейтингові списки каналів по різних параметрам. Сервіс точно знаходить тренди за ключовими словами, та видає кількість переглядів у неділю або місяць по ключовому запиту. Додаток також вираховує потенційний прибуток YouTube-каналу, проте, проаналізувавши відгуки користувачів

сервісу, було виявлено, що дані функціональні можливості сервісу працюють некоректно. Інтерфейс веб-сайту наведено на рис. 2.

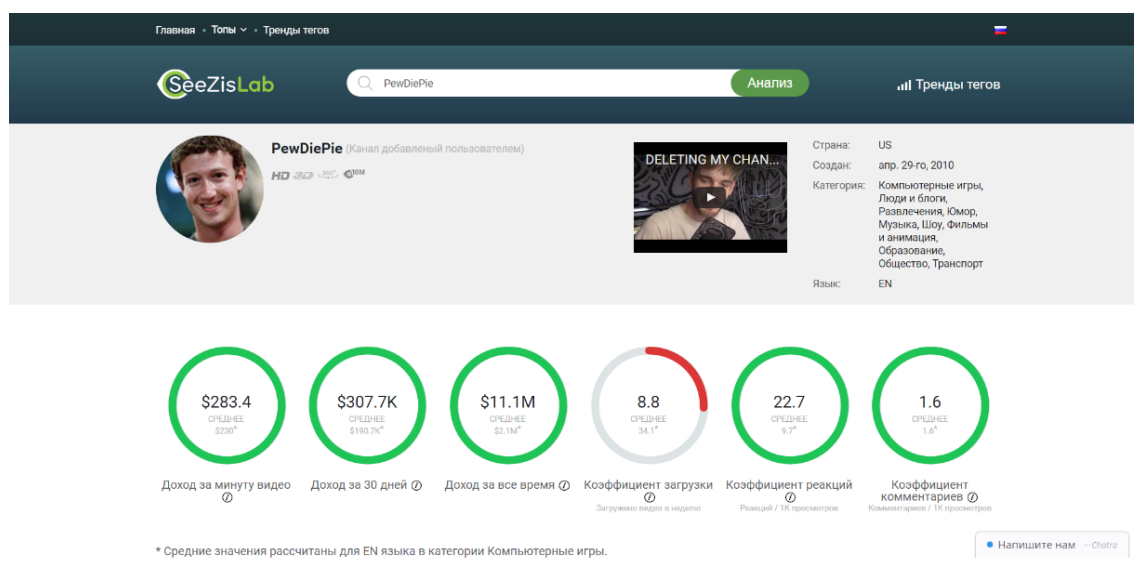


Рис. 2. Сторінка аналізу каналу на сервісі «SeeZisLab» [5]

Додаток є безкоштовним та реалізований у вигляді веб сайту. Сервіс є не стабільним та не витримує значних навантажень. Також сервіс не має функціональних можливостей для детального аналізу коментарів, аналізу їх змісту та тональності тексту повідомлень [6].

1.2.3. Додаток LiveDune

«LiveDune» – сервіс для порівняння показників залученості аккаунту у соціальних мережах. Сервіс аналізує динаміку інтересу до різних тем – вираховує популярні відео конкурентів, відстежує згадки теми по хештегам. Можливо підключити до 100 каналів одночасно. LiveDune аналізує пости та формує користувачів у тематичні рейтинги.

У додатку є стандартна статистика по охопленню каналу – кількість постів, лайки, перегляди, коментарі, усі показники демонструються у динаміці. Інтерфейс веб-сервісу наведено на рис. 3.

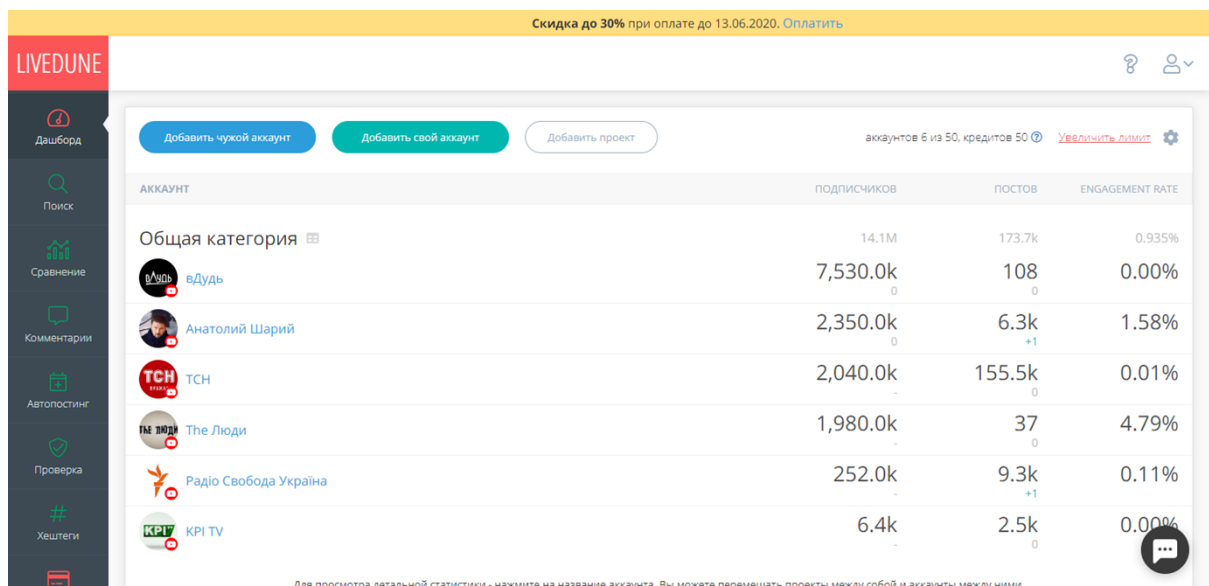


Рис. 3. Сторінка аналізу каналу на сервісі «LiveDune»

Для того щоб користувач сервісу не пропускав нові коментарі, «LiveDune» має функціональні можливості для моніторингу коментарів. У додатку можна налаштувати відправку повідомлень про нові коментарі у Telegram. Також наявне завантаження звітів у Excel. Додаток є платним та коштує від \$8 за місяць [7].

Таблиця 1

Порівняльна характеристика програмних додатків

Функціональні можливості	Popsters	SeeZisLab	LiveDune
Пошук коментарів	Немає	Немає	Немає
Відображення кількості коментарів	Є	Є	Немає
Аналіз лінгвістичної складової коментарів	Немає	Немає	Немає
Наявність безкоштовної версії	Немає	Є	Немає
Робота додатку	Стабільна	Нестабільна	Стабільна

Аналіз сторінок інших соціальних мереж окрім YouTube	Є	Немає	Є
--	---	-------	---

Отже, можемо бачити, що жоден із оглянутих сервісів для аналізу YouTube-каналів не мають функціональних можливостей для аналізу змісту текстової частини коментарів.

1.3. Актуальність розробки веб-сервісу для аналізу коментарів на YouTube

Метою розробки сервісу веб-сервісу для аналізу коментарів на YouTube є створення додатку для збору інформації та аналізу активності користувачів, включаючи аналіз змісту коментарів.

Сьогодні не існує інструмента для повноцінного аналізу коментарів на сервісі YouTube. Коментарі – це великий та важливий пласт даних, який зберігає у собі інформацію про самих залучених, а отже найважливіших глядачів. За допомогою даного сервісу, автори каналів зможуть отримувати повноцінний зворотній зв'язок від глядачів відео. Отримавши у своє розпорядження такий інструмент, як сервіс для аналізу коментарів, блогери зможуть краще спілкуватися із своєю аудиторією, оскільки у них буде можливість бачити більш цілісну картину повідомлень у коментарях.

Більшість проаналізованих вище сервісів, використовують лише статистику кількості коментарів та інколи динаміку їх росту, але жоден із відомих сервісів не використовує синтаксичний аналіз коментарів. Також більшість відомих сервісів не мають взагалі функцій пошуку коментарів, яка є дуже корисною для звичайних користувачів YouTube.

1.4. Загальні вимоги до системи

Не дивлячись на те, що у середньому більше половини переглядів відео на YouTube надходять із мобільних пристроїв, доречніше реалізувати систему у вигляді веб-додатку, оскільки функції даного сервісу знадобляться у нагоді більшості користувачам браузерної версії YouTube.

Розроблене програмне забезпечення для аналізу коментарів на YouTube повинно відповідати таким вимогам:

- забезпечення авторизації користувачів;
- можливість надавати відео для аналізу за допомогою посилання на YouTube;
- можливість переглянути коментарі у вигляді списку та їх загальну кількість, список повинен відображати інформацію про:
 - текст повідомлення;
 - час створення коментарю;
 - кількість лайків коментарю;
 - назву каналу автора коментарю;
 - зображення автора каналу коментарю;
- можливість пошуку коментарів за входженням текстового рядка у відео, яке задане користувачем;
- можливість сортування коментарів за: кількістю лайків, кількістю відповідей на коментар, датою створення коментарю;
- можливість відображення тональності тексту коментарю [8];
- забезпечення візуалізації статистики обраного відео, а саме таких графіків:
 - лінійної діаграми кількості коментарів за добу;
 - лінійної діаграми відношення кількості позитивних коментарів до негативних;
 - кругової діаграми відношення кількості позитивних коментарів до негативних;

- кругової діаграми відношення кількості відповідей до звичайних коментарів;
- гістограми середньої кількості коментарів за день тижня;
- кругової діаграми розподілу коментарів за мовою написання;
- сервіс має використовувати YouTube Data API v3 для збору коментарів;
- забезпечення можливості взаємодії клієнта із сервером за допомогою власного API;
- підтримка усіх сучасних браузерів, а саме:
 - Google Chrome (від версії 50.0);
 - Firefox (від версії 45);
 - Microsoft Edge (від версії 44);
 - Safari (від версії 9);
 - Opera (від версії 40);
- сервіс має працювати у нормальному режимі при тисячі одночасно активних сеансах;
- сервіс має бути доступним 24 години на добу;

У сучасному світі є потреба у безпеці використання веб-додатків. Використовуючи додаток, користувач хоче бути впевненим, що його дані не потраплять у загальний доступ або у руки зломисників. Тому, у ході роботи були сформульовані вимоги до безпеки системи, а саме:

- відсутність можливості JavaScript-ін'єкцій у додатку;
- авторизація користувача за допомогою Google OAuth 2.0 API;
- захист від ін'єкцій до бази даних.

1.5. Висновки до розділу

Було проаналізовано програмні додатки для аналізу відео на YouTube. Оглянуті сервіси не мають потрібних функціональних можливостей для синтаксичного аналізу коментарів.

В результаті цього аналізу можна зробити висновок, що існує потреба у створенні веб-сервісу для аналізу статистики активності та синтаксичного аналізу коментарів на YouTube, який вирішує основні задачі:

- лінгвістичний аналіз коментарів для знаходження:
 - тональності коментарів;
 - мови написання коментарів;
- візуалізація даних про коментарі;
- пошук коментарів за заданим користувачем рядком;
- фільтрація коментарів за кількістю лайків та датою створення коментаря.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Обґрунтування вибору фреймворку для програмування серверної частини

Оскільки розроблюване програмне забезпечення має бути представлене у вигляді веб-застосунку, було проведено огляд популярних фреймворків для програмування серверної частини веб-додатків.

Огляд фреймворку Fastify.js

Fastify.js – фреймворк для розробки веб-серверів на Node.js. Даний фреймворк ефективно обробляє ресурси сервера, має гарну роботу під навантаженням, проте не має широкого набору готових додатків як у конкурентів. Даний інструмент має схожий інтерфейс із найпопулярнішими фреймворками для розробки веб-серверів, таких як Express, Koa та Hapi [9].

Node.js – програмна платформа що перетворює JavaScript з вузькоспеціалізованої скриптової мови програмування у мову загального призначення [10].

Переваги Fastify.js:

- висока швидкодія фреймворку та платформи Node.js;
- єдина мова на клієнті та на сервері;
- асинхронний ввід та вивід у Node.js.

Недоліки Fastify.js:

- відсутність багатопоточності у JavaScript;
- мала кількість готових додатків до фреймворку.

Огляд фреймворку Flask

Flask – фреймворк для створення веб-додатків на мові програмування Python [11]. Фреймворк є мінімалістичним каркасом веб-застосунків який надає базові можливості для створення додатків [12].

Переваги Flask:

- низький поріг входження;
- мінімалістичний інтерфейс;

- наявність інтегрованих інструментів для тестування та налагодження;
- наявність великої кількості бібліотек на Python.

Недоліки Flask:

- відсутність асинхронності;
- не підходить для розробки великих додатків.

Огляд фреймворку Laravel

Laravel – фреймворк для створення веб-додатків на мові програмування PHP. Фреймворк призначений для розробки із використанням архітектурної моделі MVC (Model View Controller) [13].

Переваги Laravel:

- наявність доступної документації;
- низький поріг входження.

Недоліки Laravel:

- низька швидкодія;
- непослідовний синтаксис мови програмування PHP.

Огляд фреймворку Ruby on Rails

Ruby on Rails – фреймворк написаний на мові програмування Ruby, реалізує архітектурний шаблон MVC для веб-додатків, а також забезпечує інтеграцію із веб-сервером та сервером бази даних [14].

Переваги Ruby on Rails:

- висока швидкість розробки проєктів;
- сильна масштабованість додатку.

Недоліки Ruby on Rails:

- низька швидкодія;
- ціна за хостинг серверу на Ruby on Rails вища за середню по ринку;
- недостатня кількість документації;
- має погану сумісність із Windows.

Обраний фреймворк для програмування серверної частини

Проаналізувавши вимоги для створення програмного забезпечення серверної частини веб-додатку, для розробки було обрано фреймворк Fastify.js.

Fastify.js має високу швидкодію, працює на платформі Node.js та написаний на JavaScript, що дозволяє використовувати одну й ту ж мову програмування на серверній та клієнтській частині. За допомогою Fastify.js буде створено основний API для взаємодії із клієнтом.

2.2. Обґрунтування вибору фреймворку для програмування клієнтської частини

Оскільки розроблюване програмне забезпечення має бути представлене у вигляді веб-застосунку, було проведено огляд популярних фреймворків для програмування клієнтської частини веб-додатків, а саме: React, Vue.js та Angular (у даному розділі описується Angular версії 2 та вище).

Огляд фреймворку React

React – фреймворк для розробки користувацьких інтерфейсів на мові програмування JavaScript. React розроблений командою з компанії Facebook. React має більшу кількість готових компонентів та додатків за своїх конкурентів, оскільки є найбільш популярним серед розробників [15].

Переваги React:

- велика кількість готових компонентів;
- використовує розширення синтаксису JavaScript – JSX, що спрощує написання компонентів;
- легка міграція між версіями;
- фреймворк використовує віртуальний DOM.

Недоліки React:

- велика вага фреймворку у порівнянні із конкурентами.

Огляд фреймворку Angular

Angular – платформа для розробки веб-додатків на мові програмування TypeScript. Angular розроблений командою з компанії Google. Angular має високий поріг входження, адже використовує мову програмування TypeScript. Платформа використовує шаблон програмування Model-View-ViewModel, що спрощує роботу розробників із зв'язаним кодом [16].

Переваги Angular:

- TypeScript дозволяє стежити за чистотою коду у проєкті та мінімізує можливість помилок при розробці;
- велика кількість готових компонентів.

Недоліки Angular:

- проблематична міграція між версіями;
- має великий поріг входження.

Огляд фреймворку Vue.js

Vue – JavaScript-фреймворк для розробки користувацьких інтерфейсів. Vue – найновіший фреймворк із усіх описаних вище, тому він має меншу кількість готових компонентів та рішень. Фреймворк демонструє високу швидкість використовуючи віртуальний DOM. Технічно Vue реалізує шар ViewModel шаблону проєктування Model-View-ViewModel. Фреймворк з'єднує представлення та модель за допомогою двосторонньої прив'язки даних [17].

Переваги Vue:

- низький поріг входження;
- легка міграція між версіями;
- висока швидкодія;
- низька вага фреймворку у порівнянні із конкурентами;
- висока швидкість розробки;
- доступність документації;
- фреймворк використовує віртуальний DOM.

Недоліки Vue:

- не підходить для розробки дуже великих проєктів;
- невелика кількість готових компонентів у порівнянні із конкурентами.

Обраний фреймворк для програмування клієнтської частини

Проаналізувавши вимоги для створення програмного забезпечення клієнтської частини було обрано фреймворк Vue. Даний фреймворк дозволяє проєктувати клієнтські інтерфейси використовуючи мінімальні витрати часу, оскільки він має низький поріг входження та забезпечений потрібними для розробки бібліотеками та додатками.

2.3. Обґрунтування вибору системи керування базами даних

Програмне забезпечення для аналізу коментарів на YouTube має працювати із великою кількістю користувачів, зберігати інформацію про них та зібрані ними коментарі, тому використання СКБД є необхідним. Для огляду було обрано найпопулярніші SQL та NoSQL системи керування базами даних, а саме: MySQL, PostgreSQL, MongoDB та Redis.

Огляд СКБД MySQL

MySQL – реляційна СКБД. Розробку та підтримку MySQL забезпечує компанія Oracle. MySQL є кросплатформною системою керування базами даних та має дистрибутиви під різноманітні ОС, у тому числі версії під Linux, Windows та MacOS [18].

Переваги MySQL:

- є простою у встановленні та використанні;
- має зрозумілу та структуровану документацію;
- має велику кількість вбудованих функцій для забезпечення безпеки даних.

Недоліки MySQL:

- недостатня надійність;

- низька швидкість розробки;
- складна конфігурація СКБД.

Огляд СКБД PostgreSQL

PostgreSQL – об’єктно-реляційна СКБД. Оскільки PostgreSQL є вільним програмним забезпеченням, то розробка даної СКБД не контролюється певною компанією, а сирцевий код серверу PostgreSQL можна знайти у вільному доступі. Основною перевагою даної СКБД перед іншими SQL-базами даних є те, що PostgreSQL є об’єктно-реляційною, що надає їй унікального функціоналу, який відсутній у популярних реляційних СКБД [19].

Переваги PostgreSQL:

- є безкоштовною;
- має підтримку JSON;
- дозволяє створювати нові типи даних;
- висока швидкодія.

Недоліки PostgreSQL:

- відсутній вбудований шардинг;
- складна конфігурація СКБД.

Огляд СКБД MongoDB

MongoDB – документоорієнтована СКБД із відкритим вихідним кодом, підтримується розробником MongoDB Inc. MongoDB є нереляційною, або NoSQL базою даних, це означає що дана СКБД не підтримує SQL, а має власну мову запитів [20].

Переваги MongoDB:

- використовує формат документів BSON (Binary JSON), що спрощує розробку на мові програмування JavaScript;
- може зберігати дані із будь-якою структурою;
- сильна масштабованість;
- має структуровану та доступну документацію.

Недоліки MongoDB:

- працює повільніше на структурованих даних ніж реляційні СКБД;
- використовує власну мову запитів, яка не сумісна із SQL.

Огляд СКБД Redis

Redis – резидентна СКБД із відкритим вихідним кодом, підтримується розробником RedisLabs. Дана система керування базами даних відноситься до класу NoSQL та працює із структурами ключ-значення. В більшості випадків дана СКБД використовується для кешування даних [21].

Переваги Redis:

- має низький поріг входження;
- має структуровану та доступну документацію;
- має високу швидкодію.

Недоліки Redis:

- об'єм даних який може зберігати СКБД обмежений оперативною пам'яттю серверу.

Обрана СКБД

Отже, проаналізувавши усі вищеописані СКБД було обрано нереляційну базу даних MongoDB. Через простоту конфігурації та представлення документів у форматі BSON об'єктів, дана СКБД є найкращим варіантом для проєктування та значно знизить час розробки веб-сервісу.

2.4. Висновки до розділу

У даному розділі було обрано фреймворки для розробки серверної та клієнтської частини веб-додатку та систему керування СКБД. Серед технологій було розглянуто фреймворки для розробки серверної частини: Fastify.js, Flask, Laravel, Ruby on Rails; фреймворки для розробки клієнтської частини: React, Angular, Vue; системи керування базами даних: MySQL, PostgreSQL, MongoDB. Було перелічено їх переваги та недоліки.

Серед фреймворків для програмування серверної частини було обрано Fastify.js та Flask. Fastify.js було обрано через швидкість фреймворку та роботу на платформі Node.js, що надасть змогу розробляти серверну та клієнтську частини на одній мові програмування.

Серед фреймворків для програмування клієнтської частини було обрано Vue, оскільки він демонструє гарну швидкість роботи та забезпечить високу швидкість розробки веб-додатку.

Серед систем керування базами даних було обрано MongoDB, вона забезпечує високу швидкість розробки та легку взаємодію із мовою програмування JavaScript, оскільки використовує BSON-формат документів.

3. РОЗРОБЛЕННЯ ВЕБ-СЕРВІСУ ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

3.1. Загальний опис системи

Розроблюване програмне забезпечення є веб-сервісом для збирання та фільтрації коментарів на YouTube. Сервіс надає функції візуалізації коментарів у вигляді графіків та лінгвістичного аналізу.

При розробці програмного забезпечення використовувалися сучасні фреймворки та технології, що забезпечило масштабованість та велику швидкість розробки системи.

Веб-застосунок має чотири сторінки:

- Головна;
- Коментарі;
- Аналіз;
- Налаштування.

Головна сторінка

Головна сторінка відкрита за замовчуванням. Сторінка містить поле пошуку відео. Після успішного введення посилання на відео у полі пошуку користувач переадресовується на авторизацію за допомогою Google OAuth v2 [22]. Якщо користувач намагається здійснити пошук неіснуючого або недоступного відео він отримує відповідну помилку.

Сторінка коментарів

Сторінка доступна лише для авторизованих у системі користувачів. На даній сторінці користувач бачить кількість усіх зібраних коментарів та може їх переглянути. Сторінка містить вбудований плеєр YouTube, за допомогою якого можна переглянути задане відео. На сторінці є поле пошуку, за допомогою якого користувач може виконати пошук коментарів за заданим текстовим рядком. Користувач також має можливість виконувати фільтрацію за датою створення коментарю та кількістю лайків за допомогою відповідних кнопок наявних на сторінці. Користувач може перейти до сторінки аналізу коментарів за допомогою відповідної кнопки.

Сторінка аналізу

На сторінці аналізу коментарів присутні графіки які будуються за допомогою модулю ChartModule. На сторінці також наявний вбудований YouTube-плеєр для перегляду заданого відео.

Сторінка налаштування

На сторінці наявна інформація про авторизованого користувача (якщо користувач авторизований) та форма для заповнення або зміни поточного YouTube API v3 ключа.

Усі сторінки містять спільний компонент навігації, який реалізований у вигляді панелі над основним вмістом сторінки, яка містить посилання на всі доступні користувачу сторінки.

Додаток має світлу та темну тему, між якими користувач може перемикатися за допомогою відповідного компоненту у верхній панелі сайту.

Користувацький інтерфейс сайту є адаптивним, тобто відображення сайту передбачено для екранів із різним розширенням та роздільною здатністю.

У ході розробки додатку була виявлена проблема обмеженої кількості безкоштовних запитів до YouTube Data API v3 [23]. Оскільки додаток реєструє власний ключ доступу до стороннього API, то всі запити користувачів будуть здійснюватися за допомогою даного спільного ключа. Очевидно, що при значній кількості користувачів, кількість у 10 тисяч безкоштовних запитів є недостатньою для нормальної роботи додатку. Тому для забезпечення цілодобового доступу до розробленого веб-сервісу було прийнято рішення створити роль користувача «Привілейований користувач», який має змогу надати додатку власний YouTube Data API v3 ключ.

У системі передбачено два типи користувачів:

- звичайний користувач;
- привілейований користувач.

Звичайний користувач

Має доступ до сторінок аналізу та перегляду коментарів лише тих відео, у яких кількість коментарів менше двох тисяч. Користувачу потрібно ввести свій YouTube API v3 ключ на сторінці налаштувань для отримання привілейованого статусу.

Привілейований користувач

Має функціональні можливості звичайного користувача, а також не має обмеження на максимальну кількість коментарів у аналізованому відео. Користувач вважається привілейованим якщо він надав сервісу свій YouTube API v3 ключ.

3.2. Аналіз функціональних вимог до програмного додатка

Під час роботи над проєктуванням програмного забезпечення було визначено пріоритети розробки для функціональних вимог додатку.

Таблиця 2

Функціональні вимоги до веб-сервісу для аналізу коментарів на YouTube

Код вимоги	Назва вимоги	Пріоритет вимоги
01	Можливість авторизації за допомогою Google OAuth 2.0	1
02	Можливість пошуку відео за унікальним ідентифікатором	1
03	Можливість відображення коментарів із обраного відео у вигляді списку	1
04	Можливість відображення кількості коментарів у обраного відео	2
05	Можливість пошуку коментарів у відео за заданим текстовим рядком	1

06	Можливість відображення кількості коментарів знайдених за текстовим рядком	2
07	Можливість сортування коментарів за датою створення	2
08	Можливість сортування коментарів за кількістю лайків	2
09	Можливість виявлення тональності тексту коментарів	1
10	Можливість виявлення мови написання коментарів	2
11	Можливість відображення інформації про те, чи є коментар відповіддю	3
12	Можливість візуалізації статистики коментарів у вигляді графіків	1
13	Можливість переглянути основну інформацію про відео	2
14	Використання YouTube Data API v3 для збору коментарів	1

3.3. Архітектура системи

Розроблене програмне забезпечення для аналізу коментарів на YouTube поділено на такі компоненти як:

- Веб-сервер;
- Веб-клієнт;
- База даних;
- Зовнішні API.

Діаграму компонентів системи наведено на рис. 4.

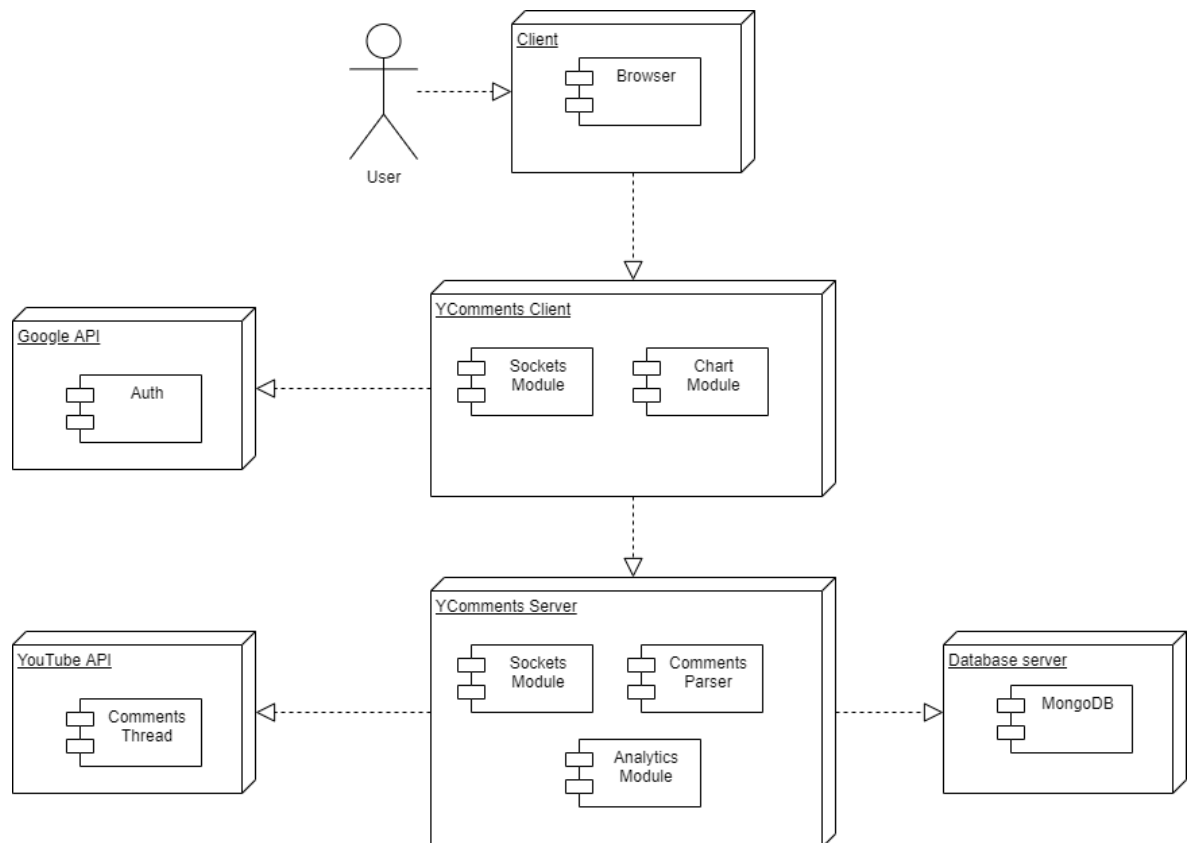


Рис. 4. Діаграма компонентів системи

3.3.1. Веб-сервер

Сервер взаємодіє із компонентами веб-клієнту (для обробки запитів користувача), серверу бази даних (для зберігання та використання зібраних коментарів та даних про користувача) та Google OAuth 2.0 API (для авторизації користувача).

Сервер складається із модулів:

- **CommentsParser** – модуль для збирання коментарів із YouTube Data API v3. Модуль рекурсивно збирає усі коментарі які є під відео за допомогою YouTube Data API v3 та зберігає їх до бази даних;
- **SocketsModule** – модуль для взаємодії із веб-клієнтом за допомогою веб-сокетів;

- **AnalyticsModule** – модуль для лінгвістичного аналізу коментарів. Модуль виконує аналіз коментарів за тональністю тексту та знаходить мову написання коментарів.

3.3.2. Веб-клієнт

Веб-клієнт дозволяє користувачу взаємодіяти із іншими компонентами програми. Клієнт надає графічний інтерфейс користувача (у вигляді веб-сторінки) та реалізовано у вигляді Single Page Application. Клієнт взаємодіє із Google OAuth 2.0 API для забезпечення авторизації користувача.

Клієнт складається із модулів:

- **SocketsModule** – модуль для взаємодії із веб-сервером за допомогою веб-сокетів;
- **ChartModule** – модуль для побудови графіків на основі даних взятих із БД.

3.3.3. Сервер бази даних

База даних додатку зберігає дані про зібрані коментарі, відео та користувачів які авторизувалися у програмі. За СКБД було обрано MongoDB.

База даних складається із таких моделей як:

- User;
- Video;
- Comment.

Схему бази даних наведено на рис. 5.

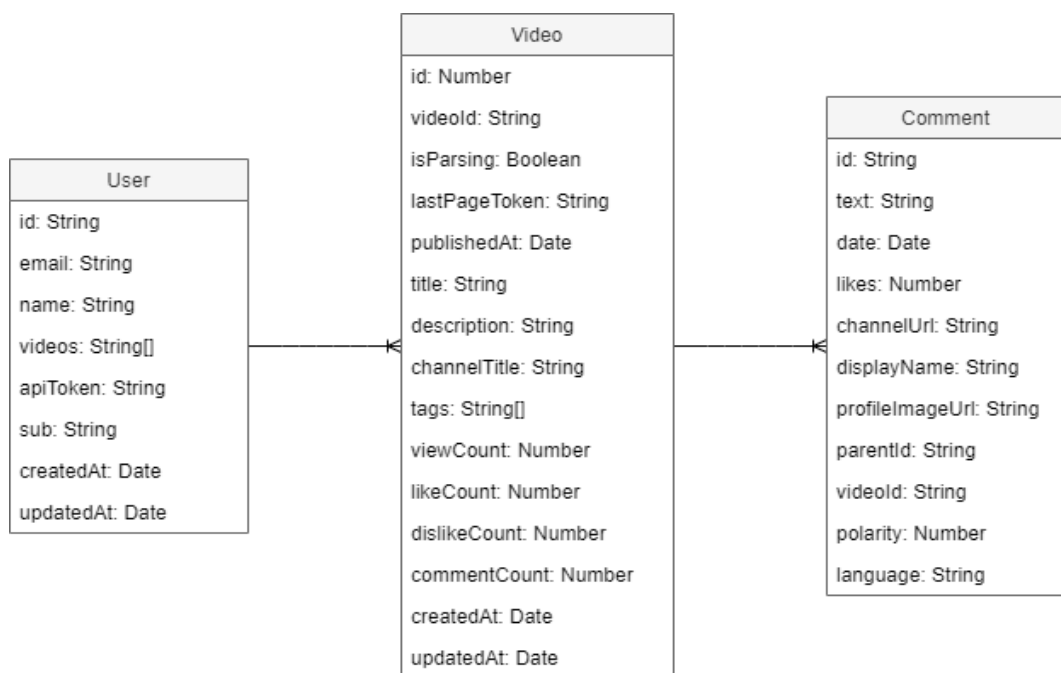


Рис. 5. Схема бази даних розроблюваного додатку

Модель User

Дана модель містить основну інформацію про користувача, а саме його унікальний ідентифікатор, електронну пошту, масив шуканих користувачем відео та YouTube API v3 ключ. Детальна інформація про назви та типи полів наведена у табл. 3.

Таблиця 3

Поля моделі User

Назва поля	Тип даних
id	Number (AUTOINCREMENT)
email	String
name	String
videos	String[]
apiToken	String
sub	String

Продовження табл. 3

createdAt	Date
updatedAt	Date

Модель Video

Модель містить у собі основну інформацію про відео, коментарі із яких були зібрані у додатку, а саме: унікальний ідентифікатор, ідентифікатор YouTube-відео, чи є коментарі із відео зібраними до кінця, токен останньої зібраної сторінки коментарів та дата останнього оновлення коментарів. Детальна інформація про назви та типи полів наведена у табл. 4.

Таблиця 4

Поля моделі Video

<i>Назва поля</i>	<i>Тип даних</i>
id	Number (AUTOINCREMENT)
videoId	String
isParsing	Boolean
lastPageToken	String
publishedAt	Date
title	String
description	String
channelTitle	String
tags	String[]
viewCount	Number
likeCount	Number
dislikeCount	Number

Продовження табл. 4

commentCount	Number
createdAt	Date
updatedAt	Date

Модель Comment

Дана модель містить інформацію про коментар взятий із YouTube-відео. Детальна інформація про назви та типи полів наведена у табл. 5.

Таблиця 5

Поля моделі Comment

<i>Назва поля</i>	<i>Тип даних</i>
id	Number (AUTOINCREMENT)
text	String
date	Date
likes	Number
channelUrl	String
displayName	String
profileImageUrl	String
parentId	String
videoId	String
polarity	Number
language	String

3.3.4. Зовнішні API

До зовнішніх API відносяться сторонні додатки та ресурси які використовуються у розроблюваному програмному забезпеченні. Такими ресурсами є: YouTube Data API v3 та Google OAuth 2.0 API.

YouTube Data API v3 використовується для збирання даних про коментарі, зокрема такі дані як:

- унікальний ідентифікатор коментарю;
- текст коментарю;
- дата створення коментарю;
- кількість лайків;
- посилання на канал автора коментарю;
- ім'я автора коментарю;
- аватар каналу автора коментарю;
- унікальний ідентифікатор батьківського коментарю.

3.4. Особливості реалізації

Додаток було реалізовано у вигляді SPA за допомогою фреймворку Vue.js. Проєкт використовує SSR реалізований за допомогою фреймворку Nuxt.js [24]. SSR забезпечує більшу швидкодію додатку навідміну від CSR та надасть підтримку індексації пошуковими роботами, що є обов'язковим при просуванні додатку у пошукових системах.

У додатку використовується фреймворк Vuetify.js за допомогою якого було реалізовано користувацький інтерфейс [25]. Інтерфейс користувача додатку є адаптивним, тобто адаптується під екрани із різним розширенням та різною роздільною здатністю.

Основними шаблонами проєктування веб-додатків є MVC (Model-View-Controller), MVP (Model-View-Presenter) та MVVM (Model-View-ViewModel). Оскільки веб-додаток для аналізу коментарів на YouTube було розроблено за допомогою фреймворку Vue.js, архітектура веб-клієнту була

реалізована у вигляді шаблону проєктування MVVM. Схема компонентів шаблону MVVM наведена на рис. 6.

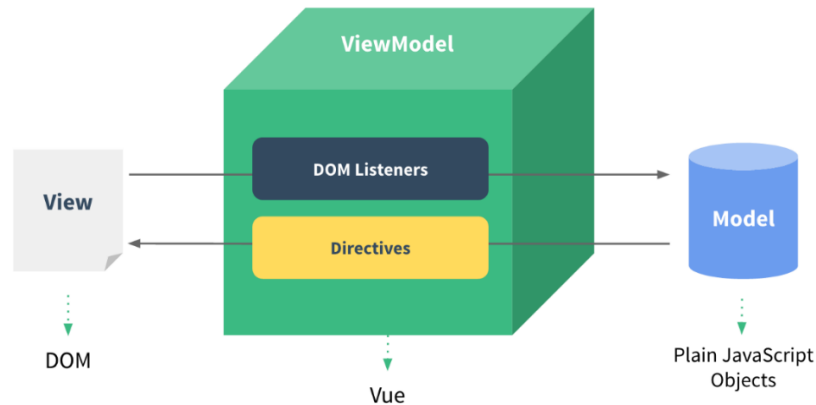


Рис. 6. Використання шаблону MVVM у фреймворку Vue.js [26]

Дана архітектура дозволяє розділити особливості реалізації на три блоки:

- Model;
- View;
- ViewModel.

Model – шар логіки, який відповідає за логіку роботи з даними та опис даних, необхідних для роботи програми. У розробленому програмному забезпеченні було створено схеми, що відповідають моделям у базі даних:

User – модель, яка має:

- унікальний ідентифікатор;
- електронну пошту;
- шукані відео;
- ключ YouTube Data API v3;
- унікальний ідентифікатор аккаунту Google;
- дата створення запису;
- дата оновлення запису.

Video – модель, яка має:

- унікальний ідентифікатор;
- ідентифікатор YouTube-відео;
- чи збираються коментарі із відео зараз;
- YouTube Data API v3 токен для останньої зібраної сторінки;
- дата публікації відео;
- назва відео;
- опис відео;
- назва YouTube-каналу автора відео;
- теги відео;
- кількість переглядів;
- кількість лайків;
- кількість дизлайків;
- кількість коментарів;
- дата створення запису;
- дата оновлення запису.

Comment – модель, яка має:

- унікальний ідентифікатор;
- текст коментарю;
- дату створення;
- кількість лайків;
- посилання на канал автора;
- ім'я автора;
- зображення автора;
- унікальний ідентифікатор батьківського коментарю;
- ідентифікатор YouTube-відео;
- значення полярності коментарю;
- мова написання коментарю.

View – шар логіки, який виступає підписником на події зміни значень властивостей чи команд, що надаються шаром ViewModel.

ViewModel – шар логіки, який є обгорткою даних із Model. Даний шар містить Model, перетворений до View, а також команди, якими може користуватися View, щоб впливати на Model. У фреймворку Vue.js, шаром ViewModel виступає об’єкт класу Vue.

Під час розробки системи було створено діаграму розгортання системи, що наведена на рис. 7.

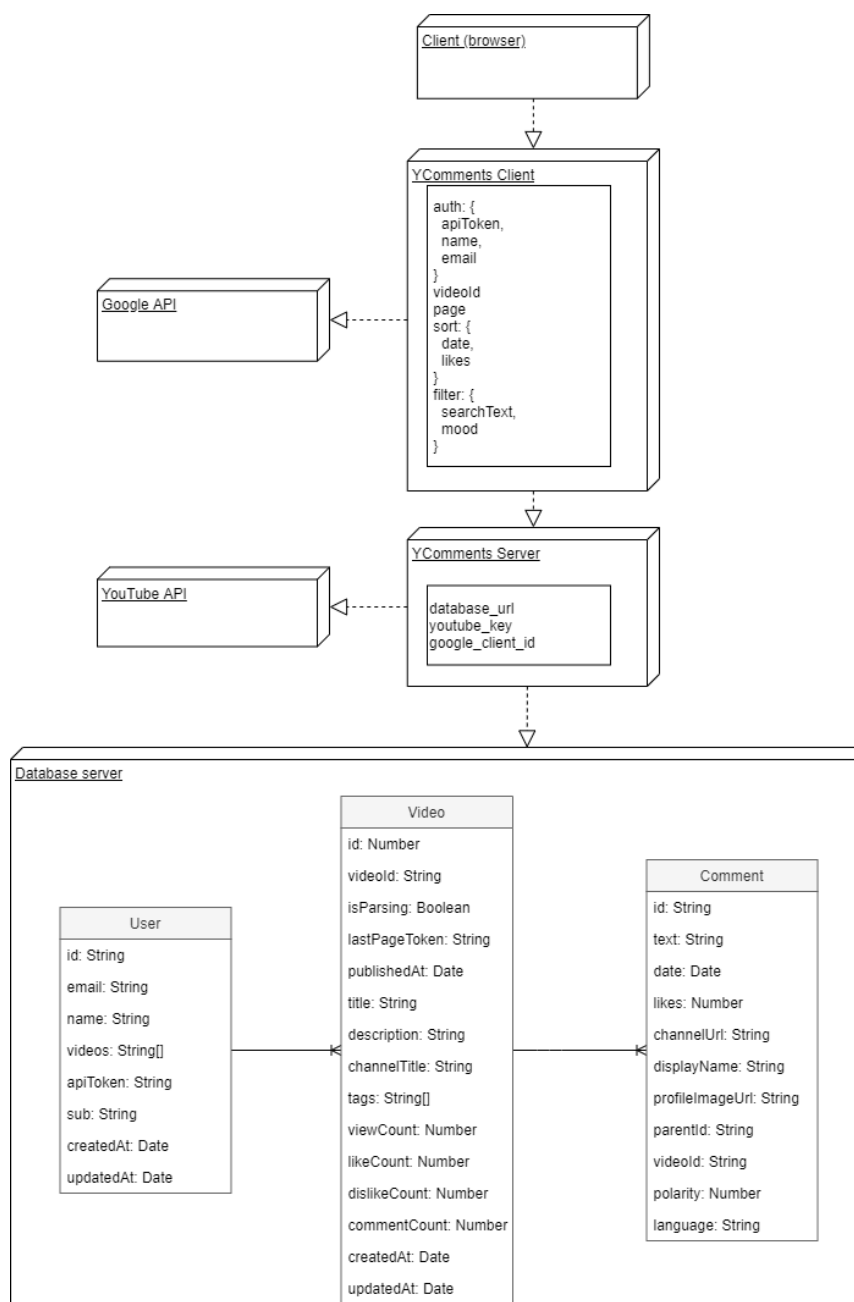


Рис. 7. Діаграма розгортання системи

Для взаємодії платформи Node.js із базою даних MongoDB було обрано застосувати фреймворк Mongoose.js. За допомогою даного фреймворку було описано схеми бази даних та здійснено підключення до СКБД.

Для здійснення лінгвістичного аналізу коментарів було написано програмний модуль AnalyticsModule. Для аналізу тональності та мови написання тексту було використано бібліотеку polyglot написану на мові програмування Python [27]. Для роботи із даною бібліотекою було написано модуль, який дозволяє використовувати потрібні функції даної Python-бібліотеки у середовищі виконання платформи Node.js за допомогою створення дочірнього процесу [28]. Даний модуль надає програмний інтерфейс, який у аргументах приймає масив речень, які потрібно проаналізувати за тональністю та надає відповідь у вигляді JSON-об'єкту.

Для побудови графіків на сторінці аналізу було написано модуль ChartModule. Для створення графіків використано фреймворк ApexCharts.js. Даний фреймворк надає зручний інтерфейс для побудови різноманітних графіків та діаграм [29]. Графіки будуються із даних отриманих клієнтом від веб-сервера у вигляді сформованому для відображення. Модуль будує такі типи графіків:

- лінійна діаграма кількості коментарів за годину;
- лінійна діаграма кількості коментарів за добу;
- лінійна діаграма відношення кількості позитивних коментарів до негативних;
- кругова діаграма відношення кількості позитивних коментарів до негативних;
- кругова діаграма відношення кількості відповідей до звичайних коментарів;
- гістограма середньої кількості коментарів за день тижня.

Для збору сервером коментарів під YouTube-відео було написано програмний модуль CommentsParser. Збір коментарів реалізується з

використанням YouTube Data API v3. Оскільки даний ресурс дозволяє отримати лише 100 коментарів за один запит, модуль CommentsParser використовує рекурсивний метод збору коментарів. Даний модуль використовує AnalyticsModule для визначення тональності коментаря та подальшого занесення даної інформації у базу даних.

Для роботи із веб-сокетами на клієнті та сервері було використано фреймворк Socket.IO. Даний фреймворк надав зручний інтерфейс для роботи із веб-сокетами для платформи Node.js. Під час запиту на пошук коментарів із відео, користувач додається у простір імен сокетів із назвою яка співпадає з унікальним ідентифікатором YouTube-відео. До даних сокетів із веб-серверу надсилається поточна інформація про зміст та кількість коментарів під час їх збору.

3.5. Висновки до розділу

У даному розділі було описано архітектуру та особливості реалізації створеного програмного забезпечення для аналізу коментарів на YouTube. Було описано та розподілено права ролей користувачів сервісу – звичайного та привілейованого користувача, описано вміст сторінок сервісу. У розділі було детально описано роботу та взаємодію між модулями CommentsParser, ChartModule, AnalyticsModule та SocketsModule. Також було описано основні фреймворки та додатки які використовуються у розроблюваному програмному забезпеченні.

У розділі було описано архітектуру системи – MVVM (“Model-View-ViewModel”) та використання її у Vue.js.

Обраною СКБД для розробки сервісу є MongoDB, для роботи із якою на веб-сервері було обрано ODM Mongoose. У розділі було надано перелік існуючих моделей у системі керування базами даних.

Для лінгвістичного аналізу коментарів було використано бібліотеку `polyglot`, яка написана на мові програмування Python. За допомогою даної бібліотеки здійснюється аналіз коментарів за тональністю та знаходження мови написання коментарів.

4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБ-СЕРВІСУ ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

4.1. Аналіз розробленого веб-додатку

У результаті розроблення програмного забезпечення для аналізу коментарів на YouTube було розроблено автоматизовану систему збору коментарів із відео із можливостями їх подальшого аналізу.

Веб-сервіс було розроблено відповідно до сформованих функціональних вимог до системи (див. підрозділ 1.4. “Загальні вимоги до системи”), та було розроблено рішення:

- реалізовано авторизацію користувачів за допомогою Google OAuth 2.0. Для реалізації авторизації було використано стандартний модуль для Nuxt.js @nuxt/auth. Оскільки під час проєктування системи було прийнято рішення;
- реалізовано можливість пошуку коментарів із YouTube-відео за його посиланням. Система отримує із посилання унікальний ідентифікатор відео та відправляє відповідний запит на розроблений веб-сервер із вказанням цього ідентифікатора у параметрах. Сервер за допомогою розробленого модулю CommentsParser отримує всі коментарі, та за допомогою розробленого модулю SocketsModule відправляє їх до веб-клієнту;
- реалізовано можливість пошуку коментарів за заданим користувачем текстовим рядком. Розроблений веб-сервер шукає коментарі із входженням отриманого у запиті текстового рядка та надсилає інформацію про них клієнту, а саме їх кількість та пагінований масив коментарів;
- реалізовано можливість сортування коментарів за кількістю лайків та датою створення коментарю;
- реалізовано можливість виявлення тональності тексту та мови написання коментарів;

- реалізовано візуалізацію даних про коментарів у вигляді графіків, а саме:
 - лінійну діаграму кількості коментарів за добу;
 - лінійну діаграму відношення кількості позитивних коментарів до негативних;
 - кругову діаграму відношення кількості позитивних коментарів до негативних;
 - кругову діаграму відношення кількості відповідей до звичайних коментарів;
 - гістограму середньої кількості коментарів за день тижня;
 - кругову діаграму розподілу коментарів за мовою написання.

Розроблене програмне забезпечення для аналізу коментарів на YouTube відповідає таким нефункціональним вимогам:

- інтерфейс користувача веб-додатку є адаптивним та підлаштовується під екрани із різним розширенням та різною роздільною здатністю;
- мінімальні версії підтримки браузером:
 - Google Chrome 50.0;
 - Firefox 45;
 - Microsoft Edge 44;
 - Safari 9;
 - Opera 40.

4.2. Тестування розробленого веб-додатку

Тестування є важливим етапом розробки програмного забезпечення, оскільки воно дозволяє відловити помилки та можливі недоліки системи під час її розробки. Тестування програмного забезпечення виконувалося на кожній ітерації розробки.

Тестування розробленого програмного забезпечення відбувалося

ручним тестуванням та тестуванням потенційними користувачами.

Для тестування клієнтської частини додатку було проведено тестування потенційними користувачами. Кожен із п'яти користувачів використовував додаток впродовж десяти хвилин. Внаслідок тестування були знайдені такі помилки як:

- некоректна обробка користувацької помилки при вводі посилання без унікального ідентифікатора YouTube-відео;
- дублювання коментарів-відповідей у списку коментарів;
- можливість NoSQL ін'єкції через поле пошуку коментаря за текстовим рядком;
- невідповідність кількості відображених коментарів та коментарів під YouTube-відео;
- неправильне відображення часу створення коментарю;
- неправильне відображення коментарів із авторами, які не мали зображення каналу.

Також було проведено ручне тестування додатку, що дало змогу знайти значну кількість помилок під час розробки графічного інтерфейсу користувача та модулю системи керування базами даних.

При тестуванні додатку було враховано та виправлено усі знайдені помилки та недоліки.

4.3. Порівняння розробки із існуючими аналогами

У першому розділі дипломного проєкту були проаналізовані три аналоги розробленого програмного додатка (табл. 1). Розглянуті продукти лише частково мають спільний функціонал із розробленим додатком для аналізу коментарів на YouTube.

Жоден із розглянутих аналогів не використовує зміст текстової частини коментарів у своєму аналізі. Розглянуті аналоги мають ширшу область застосування та не спеціалізуються лише на аналізі коментарів.

Додаток Popsters

Даний додаток має функціональні можливості для аналізу YouTube-каналів або сторінок у інших соціальних мережах. «Popsters» використовує коментарі лише для побудови графіків середньої кількості коментарів за день тижня, та графіків сумарної кількості коментарів у аналізованому періоді. Додаток є платним та не має функціональних можливостей для аналізу коментарів.

Додаток SeeZisLab

«SeeZisLab» – безкоштовний додаток для аналізу YouTube-каналів. Даний сервіс не має аналізу коментарів у своїй статистиці, додаток націлений на створення рейтингових списків YouTube-каналів за різними параметрами. Даний додаток не витримує значних навантажень, оскільки є безкоштовним. Сервіс не має функціональних можливостей для аналізу коментарів.

Додаток LiveDune

Даний сервіс орієнтований на аналіз та моніторинг акаунтів у різних соціальних мережах. Сервіс не має функціональних можливостей для аналізу коментарів на YouTube.

Розроблене програмне забезпечення

Розроблене програмне забезпечення може конкурувати із іншими системами які мають функціональні можливості для аналізу каналів на YouTube. Оскільки розроблене програмне забезпечення орієнтоване лише на аналіз та обробку коментарів, воно має функціональні можливості яких не мають додатки аналоги.

4.4. Рекомендації щодо подальшого вдосконалення

Потенційні шляхи для вдосконалення системи були виявлені під час розробки та тестування системи, а також в результаті опитування серед потенційних користувачів системи.

Було запропоновано такі вдосконалення для сервісу:

- аналіз коментарів у обраного каналу, замість аналізу окремого відео;
- вдосконалити систему пошуку коментарів;
- дозволити аналізувати відео із будь-якою кількістю коментарів та прибрати роль привілейованого користувача;
- можливість аналізувати коментарі прямих трансляцій;
- можливість автоматичного оновлення коментарів відповідно до коментарям під YouTube відео;
- додавання більшої кількості графіків;
- додати можливість аналізувати інші сервіси та соціальні мережі окрім YouTube;
- створення системи оповіщень про нові коментарі;
- можливість зберігати проаналізовані дані у вигляді Excel-таблиці чи сторінки у форматі PDF;
- можливість ставити лайк коментарю безпосередньо із додатку.

4.5. Висновки до розділу

У даному розділі було проведено аналіз розробленого програмного забезпечення для аналізу коментарів на YouTube. У розділі було описано рішення для всіх функціональні вимог додатка.

При розробці додатку застосовувалося автоматичне тестування серверної частини та тестування потенційними користувачами для всієї системи. Розробником застосовувалося ручне тестування для знаходження помилок зв'язаних із дизайном користувацького інтерфейсу та модулю системи керування базами даних. Тестування системи проводилося на кожній ітерації розробки. Усі знайдені недоліки та помилки були враховані та виправлені у наступних ітераціях розробки.

Було проведено порівняння розробленого сервісу із аналогічними додатками. Розроблений додаток для аналізу коментарів на YouTube може

конкурувати з наявними на ринку рішеннями, оскільки розглянуті аналоги мають ширшу область застосування та не мають функцій для аналізу.

Під час тестування було проведено опитування серед потенційних користувачів системи, згідно якого було визначено та описано рекомендації щодо подальшого вдосконалення розробленої системи.

ВИСНОВКИ

Метою даного дипломного проєкту було створення програмного забезпечення, призначеного для спрощення та автоматизації аналізу, пошуку, фільтрації, візуалізації статистики та лінгвістичного аналізу зібраних коментарів із відео на YouTube заданого користувачем.

Під час роботи було проведено огляд існуючих рішень для аналізу YouTube-каналів представлених на ринку та сформовано вимоги до розроблюваної системи у вигляді функціональних вимог.

Було проведено порівняння розроблюваних програмних засобів для створення веб-додатку для аналізу коментарів на YouTube. Програмне забезпечення для аналізу коментарів на YouTube розроблено із використанням сучасних інструментів розробки. Для створення серверної частини додатку було обрано фреймворк Fastify.js із платформою виконання Node.js, а для клієнтської частини – фреймворк Vue.js. Обраною системою керування базами даних було обрано MongoDB.

Було створено програмне забезпечення для аналізу коментарів на YouTube, яке відповідає описаним функціональним вимогам та надає такі можливості:

- можливість пошуку відео за посиланням на YouTube;
- можливість переглянути усі коментарі із обраного відео;
- можливість відсортувати знайдені коментарі за різними параметрами;
- можливість відфільтрувати коментарі за різними параметрами;
- можливість виявити мову написання коментарів;
- візуалізація статистичних даних про коментарі у вигляді графіків та діаграм.

Було проведено тестування системи на потенційних користувачах та методом ручного тестування. Було знайдено та описано потенційні шляхи для подальшого вдосконалення та розширення системи.

Розроблена система дозволяє блогерам та користувачам YouTube аналізувати коментарі із відео. Даний сервіс, на відміну від сервісів-конкурентів, використовує лінгвістичний аналіз тексту повідомлень та може демонструвати візуалізовані дані для швидкої демонстрації інформації про коментарі.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Digital 2020: 3.8 billion people use social media : [Електронний ресурс]. Режим доступу: <https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media>
2. Google Connected Consumer Study 2017 : [Електронний ресурс]. Режим доступу: <https://ukraine.googleblog.com/2017/10/google-connected-consumer-study-2017.html>
3. Social networks ranked by number of users : [Електронний ресурс]. Режим доступу: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>
4. Статистика и аналитика контента сообществ в социальных сетях «Popsters» : [Електронний ресурс]. Режим доступу: <https://popsters.ru/>
5. SeeZisLab — обзор сервиса | Startpack : [Електронний ресурс]. Режим доступу: <https://startpack.ru/application/seezislab>
6. «SeeZisLab» – Track YouTube Channel Statistics and Analytics : [Електронний ресурс]. Режим доступу: <https://seezislab.com/>
7. Аналитика в соцсетях : [Електронний ресурс]. Режим доступу: <https://livedune.ru/>
8. Sentiment analysis : [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Sentiment_analysis
9. Fastify, Fast and low overhead web framework, for Node.js : [Електронний ресурс]. Режим доступу: <https://www.fastify.io/>
10. Янг, Алекс. Node.js в действии [Текст] / Алекс Янг, Брэдли Мек, Майк Кантелон. — СПб. : Питер, 2015. — 448 с.
11. Python Cookbook [Text] / D. Beazley, В. К. Jones. — 3rd edition. — Sebastopol : O'Reilly Media, 2013. — 688 p.
12. Welcome to Flask : [Електронний ресурс]. Режим доступу: <https://flask.palletsprojects.com/en/1.1.x/>

13. Laravel : [Электронный ресурс]. Режим доступа: <https://en.wikipedia.org/wiki/Laravel>
14. Rusrails: Ruby on Rails по-русски : [Электронный ресурс]. Режим доступа: <http://rusrails.ru/>
15. React – JavaScript-библиотека для создания пользовательских интерфейсов : [Электронный ресурс]. Режим доступа: <https://ru.reactjs.org/>
16. Angular – введение и начало работы : [Электронный ресурс]. Режим доступа: <https://metanit.com/web/angular2/1.1.php>
17. Vue.js in Action [Text] / E. Hanchett, B. Listwon. — Shelter Island : Manning, 2018. — 304 p.
18. Welcome to MySQL : [Электронный ресурс]. Режим доступа: <https://www.mysql.com/>
19. PostgreSQL: Up and Running [Text] / R. Obe, L. Hsu. — Sebastopol : O'Reilly Media, 2012. — 164 p.
20. MongoDB in Action [Text] / K. Banker, P. Bakum, S. Verch, D. Garreett, T. Hawkins. — 2nd edition. — Shelter Island : Manning, 2016. — 456 p.
21. Redis : [Электронный ресурс]. Режим доступа: <https://en.wikipedia.org/wiki/Redis>
22. Using OAuth 2.0 to Access Google APIs : [Электронный ресурс]. Режим доступа: <https://developers.google.com/identity/protocols/oauth2>
23. CommentThreads | YouTube Data API | Google Developers : [Электронный ресурс]. Режим доступа: <https://developers.google.com/youtube/v3/docs/commentThreads/list>
24. Nuxt.js — Универсальные приложения на Vue.js : [Электронный ресурс]. Режим доступа: <https://ru.nuxtjs.org/>
25. Vue Material Design Component Framework — Vuetify.js : [Электронный ресурс]. Режим доступа: <https://vuetifyjs.com/>
26. Vue.js. Getting Started : [Электронный ресурс]. Режим доступа: <https://012.vuejs.org/guide/>

27. Polyglot's documentation : [Электронный ресурс]. Режим доступа:
<https://polyglot.readthedocs.io/en/latest/>
28. Child Process | Node.js v12.16.3. Documentation : [Электронный ресурс].
Режим доступа: https://nodejs.org/docs/latest-v12.x/api/child_process.html#child_process_child_process_exec_command_options_callback
29. Apexcharts documentation : [Электронный ресурс]. Режим доступа:
<https://apexcharts.com/docs/>

ДОДАТКИ

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«__» _____ 2019 р.

ВЕБ-СЕРВІС ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

Програма та методика тестування

ДП.045440-04-51

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Юрій БРУСЕНЦОВ

ЗМІСТ

1. Об'єкт випробувань.....	3
2. Мета тестування.....	3
3. Методи тестування.....	3
4. Засоби та порядок тестування.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Веб-сервіс для аналізу коментарів на YouTube, що є веб-сайтом, створеним з використанням фреймворку Fastify.js на основі платформи Node.js для серверної частини та Vue.js для клієнтської частини.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- 1) коректна робота елементів сторінок веб-сервісу;
- 2) відповідність реалізованих функціональних можливостей веб-сервісу до описаних;
- 3) коректність роботи модуля для збору коментарів з відео-хостингу YouTube;
- 4) забезпечення належного рівня безпеки даних;
- 5) зручність роботи з веб-сервісом;

3. МЕТОДИ ТЕСТУВАННЯ

Функціональне тестування було проведено з використанням методу White Box Testing. При використанні даного методу перевіряється як код, так і коректність роботи тестованого програмного застосунку. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- 1) Critical path test (тестування критичного шляху) для проведення функціонального тестування;
- 2) Load testing (навантажувальне тестування) та Stress testing (стресс-тестування) для тестування продуктивності програмного забезпечення;
- 3) тестування інтерфейсу.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Коректність роботи веб-сервісу перевіряється з використанням таких методів та засобів:

- 1) введенням некоректних значень в поля, що можна редагувати;
- 2) ручною перевіркою розроблених функціональних можливостей до тих, що були описані;
- 3) тестування при максимальному навантаженні за допомогою утиліти Apache JMeter;
- 4) тестування стабільності роботи за допомогою утиліти Apache JMeter;
- 5) тестування зручності використання інтерфейсу опитуванням потенційних користувачів;
- 6) тестування веб-сервісу в різних браузерях.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2020 р.

ВЕБ-СЕРВІС ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

Керівництво користувача

ДП.045440-05-34

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Юрій БРУСЕНЦОВ

ЗМІСТ

1. Опис структури веб-сервісу	3
2. Опис головної сторінки	4
3. Процедура авторизації користувача	6
4. Процедури фільтрації, сортування та пошуку коментарів	7
5. Перегляд візуалізованих даних про коментарі	10
6. Опис сторінки налаштувань	14

1. Опис структури веб-сервісу

Веб-сервіс для збирання та фільтрації коментарів на YouTube надає функції візуалізації коментарів у вигляді графіків та лінгвістичного аналізу.

Веб-застосунок складається з чотирьох динамічних сторінок, на яких можна обирати світлу або темну тему:

- «Головна»;
- «Коментарі»;
- «Аналіз»;
- «Налаштування».

Сторінки поєднуються між собою навігаційною панеллю.

2. Опис головної сторінки

Головна сторінка відкривається за замовчуванням (рис. 1). Для роботи з сервісом, користувач має авторизуватися. Після авторизації, сторінка відображує поле пошуку відео (рис. 2), куди користувач може ввести своє посилання на відео. Після успішного введення коректних даних, користувач перенаправляється на сторінку «Коментарі». Якщо користувач намагається здійснити пошук неіснуючого або недоступного відео він отримує відповідну помилку (рис. 3).

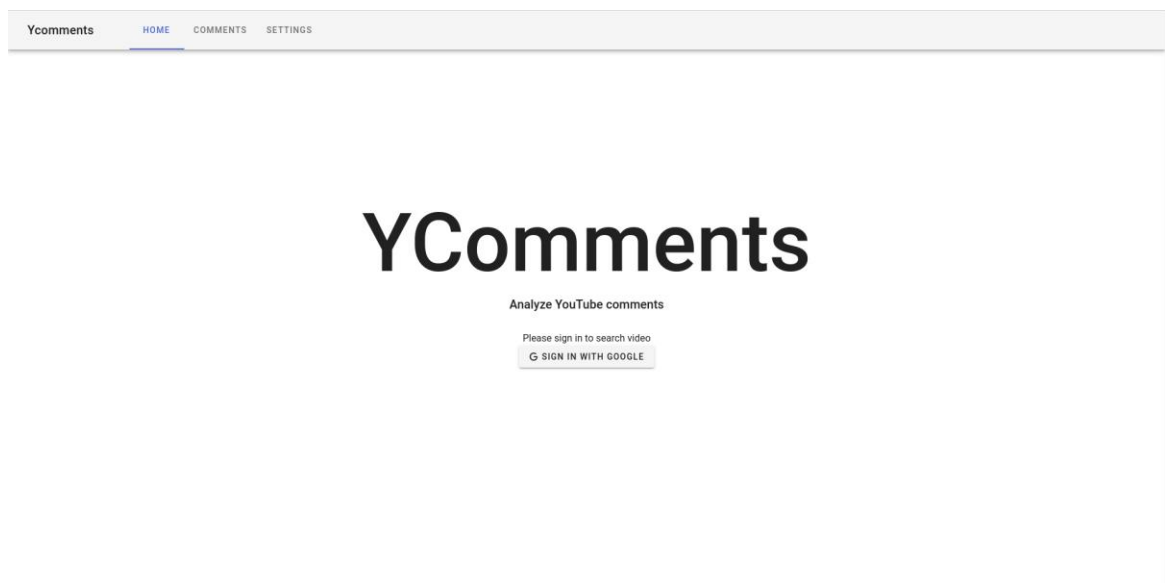


Рис. 1. Головна сторінка до авторизації



Рис. 2. Головна сторінка після авторизації

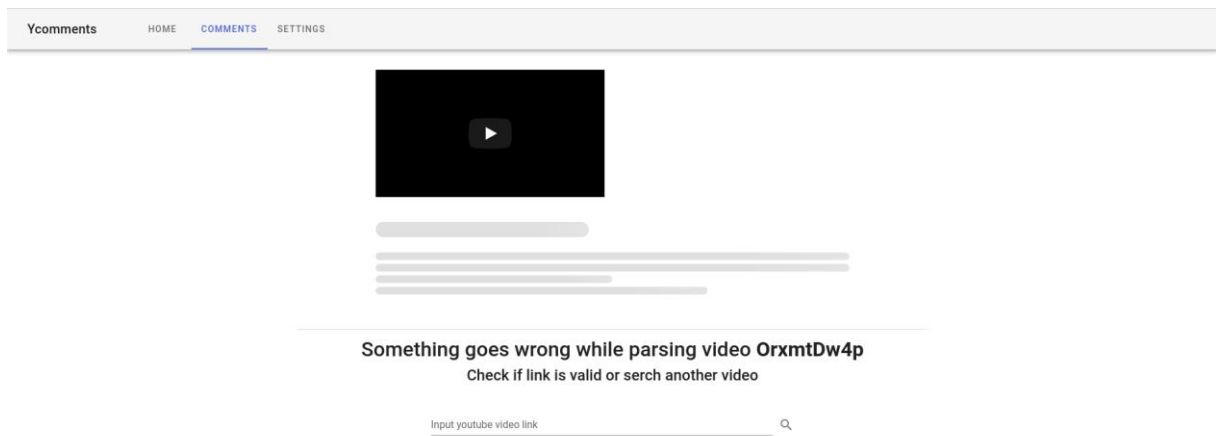


Рис. 3. Помилка при некоректному запиті

3. Процедура авторизації користувача

Оскільки авторизація користувача проводиться з використанням Google OAuth v2, то для успішної авторизації необхідно ввести логін та пароль від існуючого облікового запису Google у відповідні поля (рис. 4).

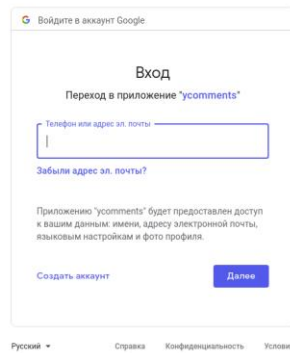


Рис. 4. Авторизація за допомогою Google OAuth v2

4. Процедури фільтрації, сортування та пошуку коментарів

Сторінка «Коментарі» (рис. 5) відображається після успішного введення посилання на відео у полі пошуку на головній сторінці. На даній сторінці користувач бачить кількість усіх зібраних коментарів та може їх переглянути. Сторінка містить вбудований плеєр YouTube, за допомогою якого можна переглянути задане відео. На сторінці є поле пошуку, за допомогою якого користувач може виконати пошук коментарів за заданим текстовим рядком (рис. 6). Користувач також має можливість виконувати фільтрацію за датою створення коментарю (рис. 7), за кількістю лайків (рис. 8) та за мовою, на якій написано коментар (рис. 9) за допомогою відповідних полів наявних на сторінці. Пошук, фільтрацію та сортування можна поєднувати в одному запиті (рис. 10).

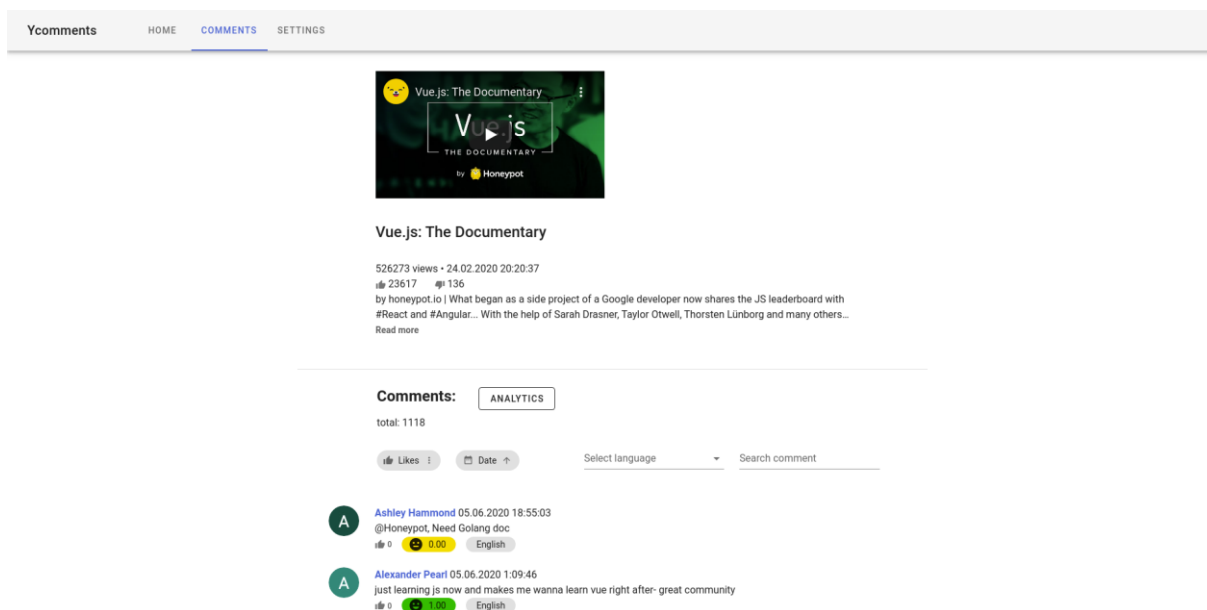


Рис. 5. Сторінка з коментарями

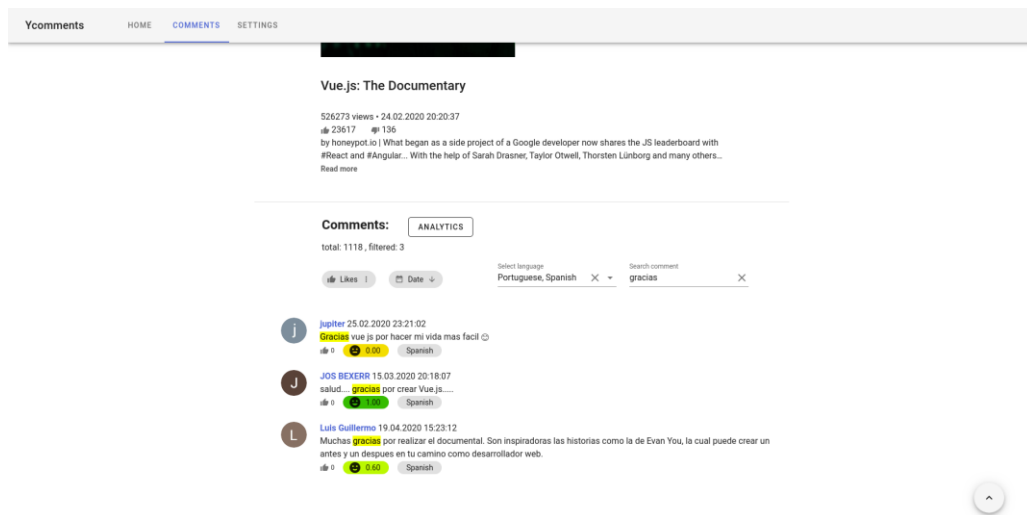


Рис. 6. Коментарі, що містять в своєму тексті заданий рядок

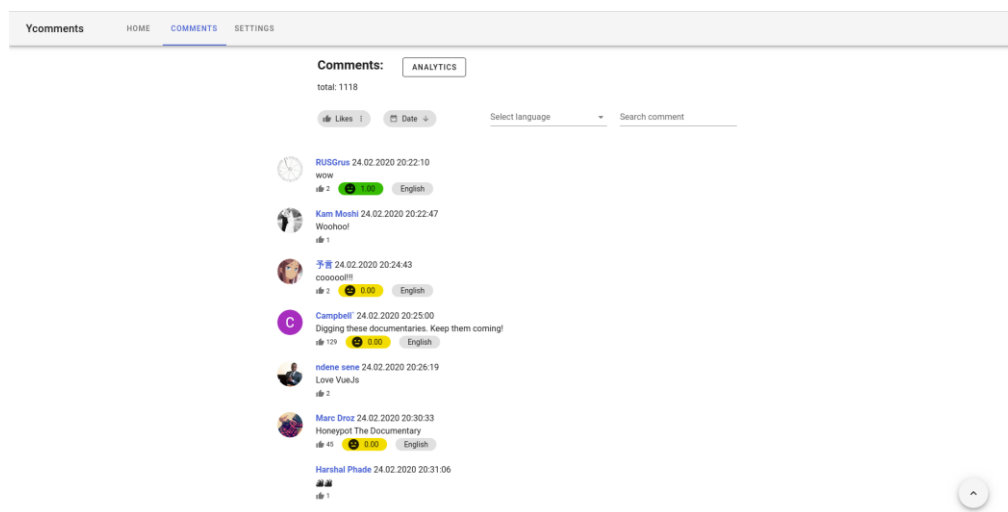


Рис. 7. Коментарі, відсортовані за датою

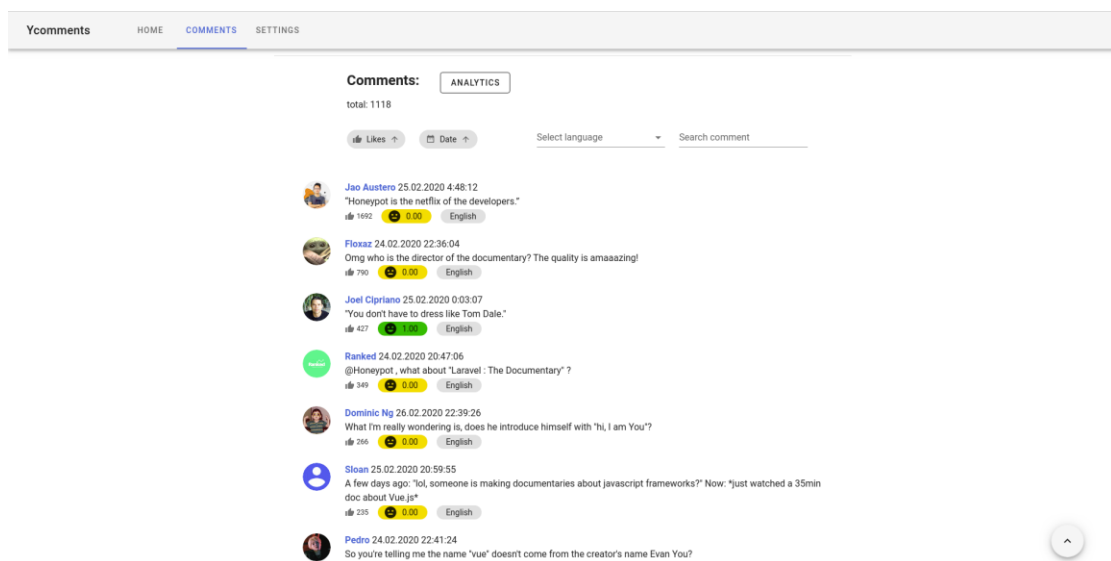


Рис. 8. Коментарі, відсортовані за кількістю лайків

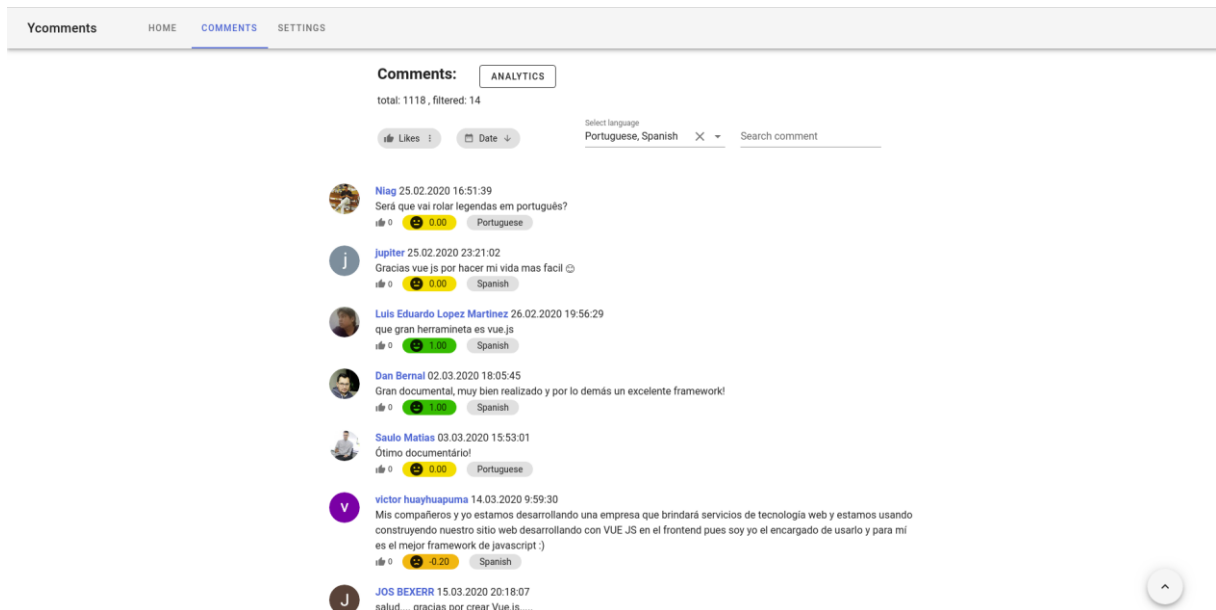


Рис. 9. Коментарі, відфільтровані за мовою

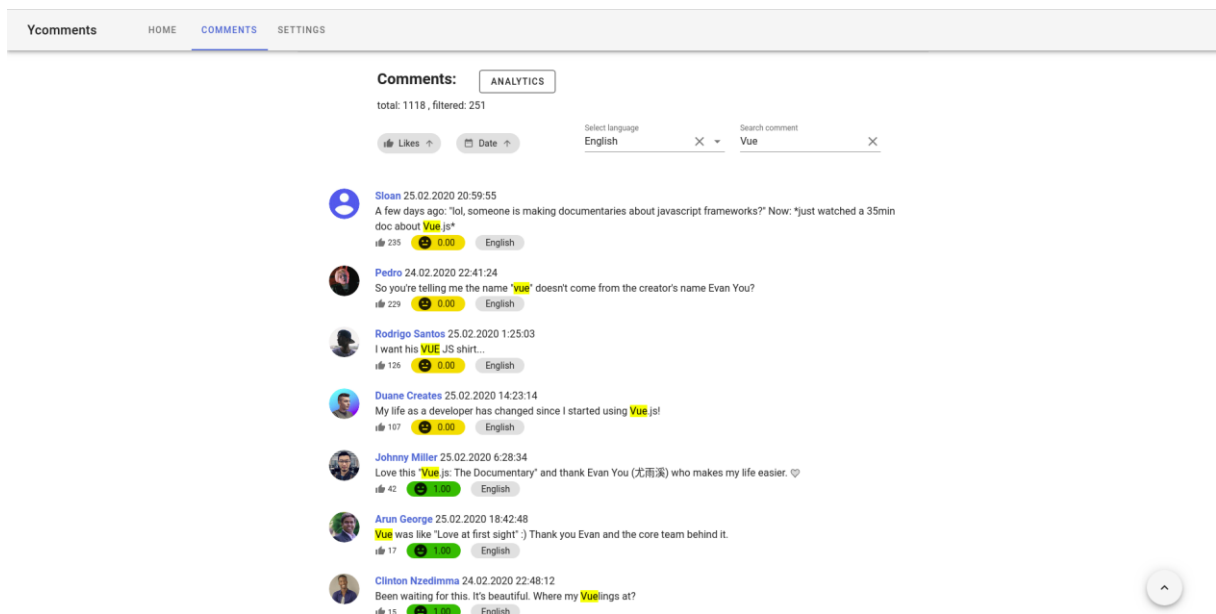


Рис. 10. Коментарі, що відсортовані за кількістю лайків, написані англійською мовою та містять в своєму тексті слово «Vue»

5. Перегляд візуалізованих даних про коментарі

Натиснувши на кнопку «ANALITYCS» (рис. 10) на сторінці «Коментарі», користувач потрапляє на сторінку аналізу, повернутися до коментарів можна, натиснувши на кнопку «COMMENTS LIST» (рис. 11). На ній відображуються графіки, що візуалізують зібрані статистичні дані про коментарі:

- «Comments per day» – лінійна діаграма кількості коментарів за добу (рис. 12);
- «Total comments polarity» – кругова діаграма відношення кількості позитивних коментарів до негативних (рис. 12);
- «Comments language distribution» – кругова діаграма відношення кількостей коментарів, написаних різними мовами (рис. 13);
- «Comments per day of week» – гістограма середньої кількості коментарів за день тижня (рис. 13);
- «Comments polarity per day» – лінійна діаграма кількості позитивних та негативних коментарів (рис. 14);
- «Replies distribution» – кругова діаграма відношення кількості коментарів до відповідей на коментарі (рис. 15).

На сторінці також наявний вбудований YouTube-плеєр для перегляду заданого відео.

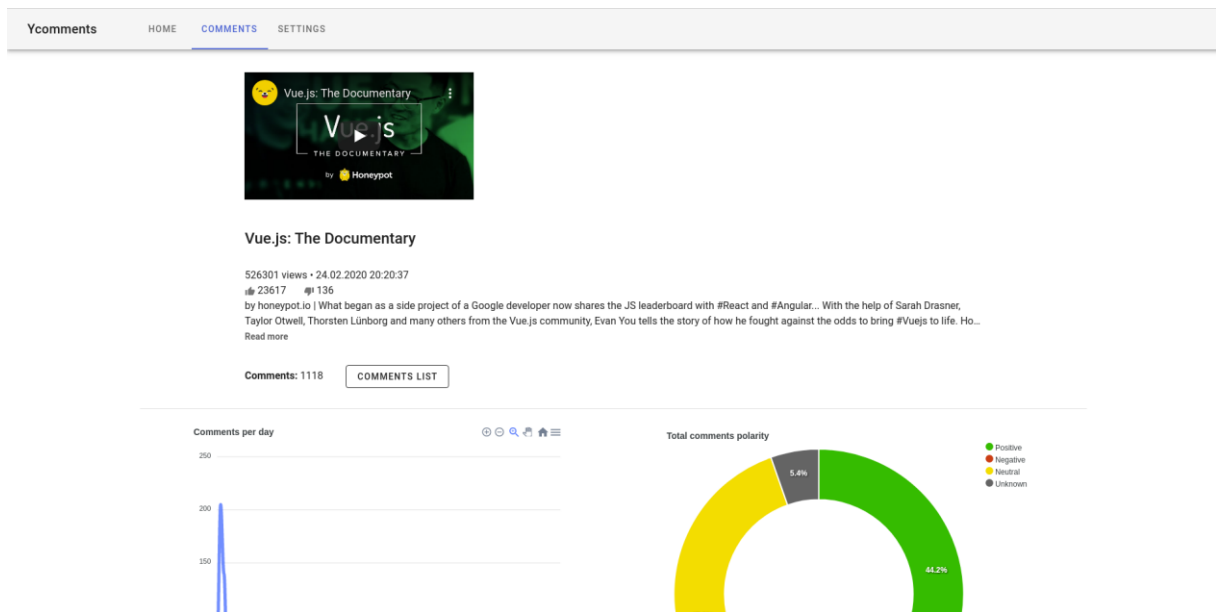


Рис. 11. Сторінка з аналізом коментарів

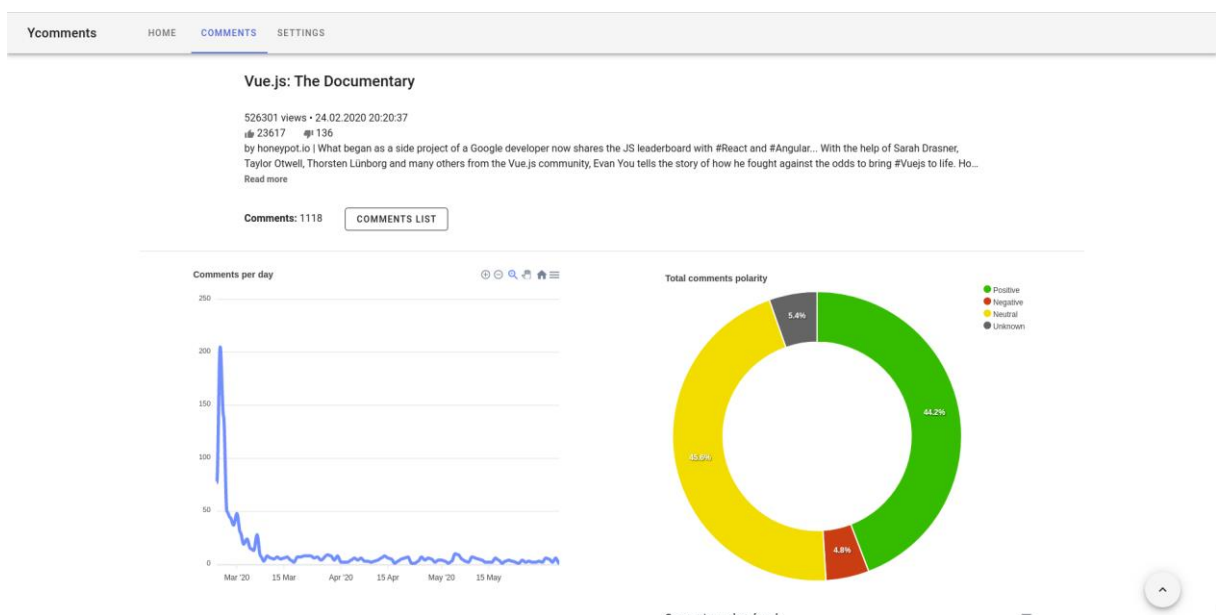


Рис. 12. Графіки «Comments per day» та «Total comments polarity»

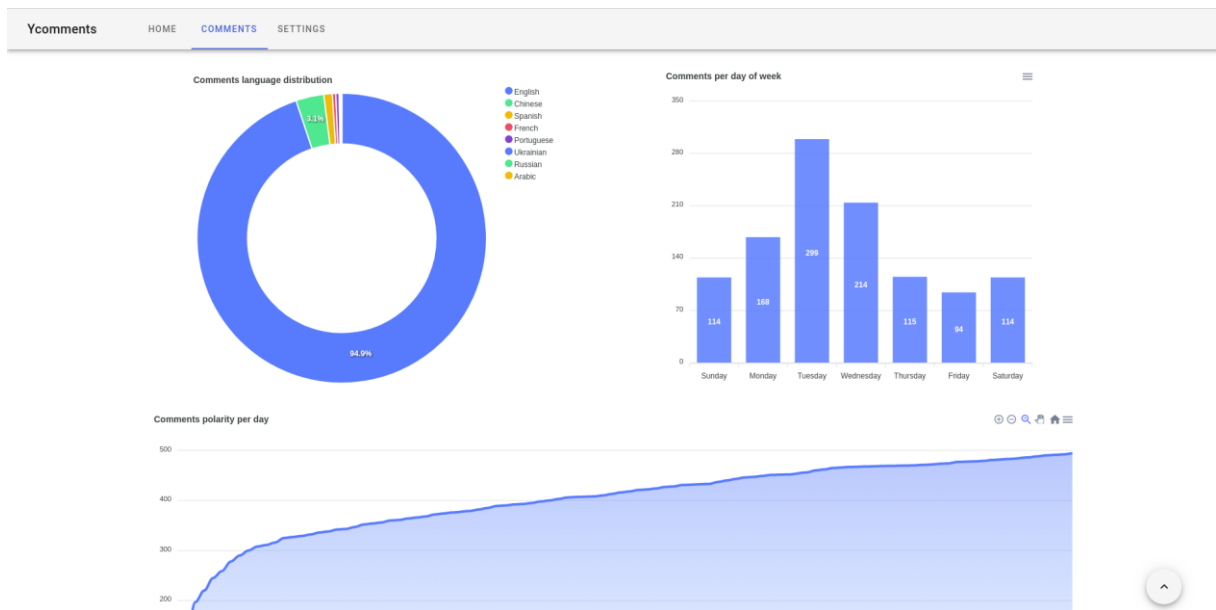


Рис. 13. Графіки «Comments language distribution» та «Comments per day of week»

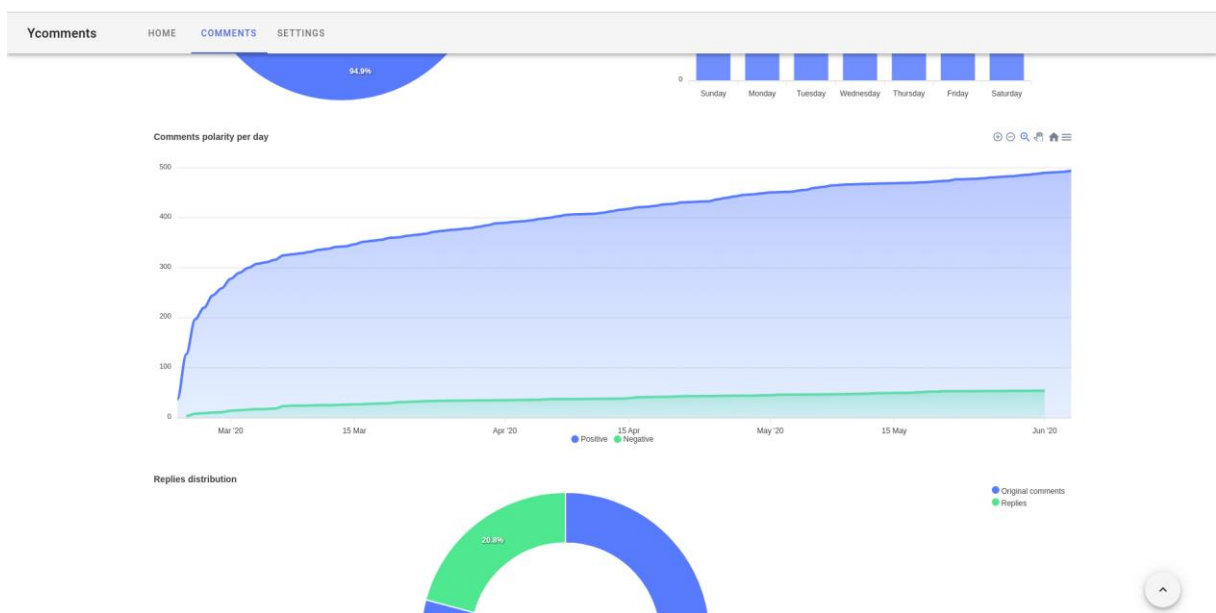


Рис. 14. Графік «Comments polarity per day»

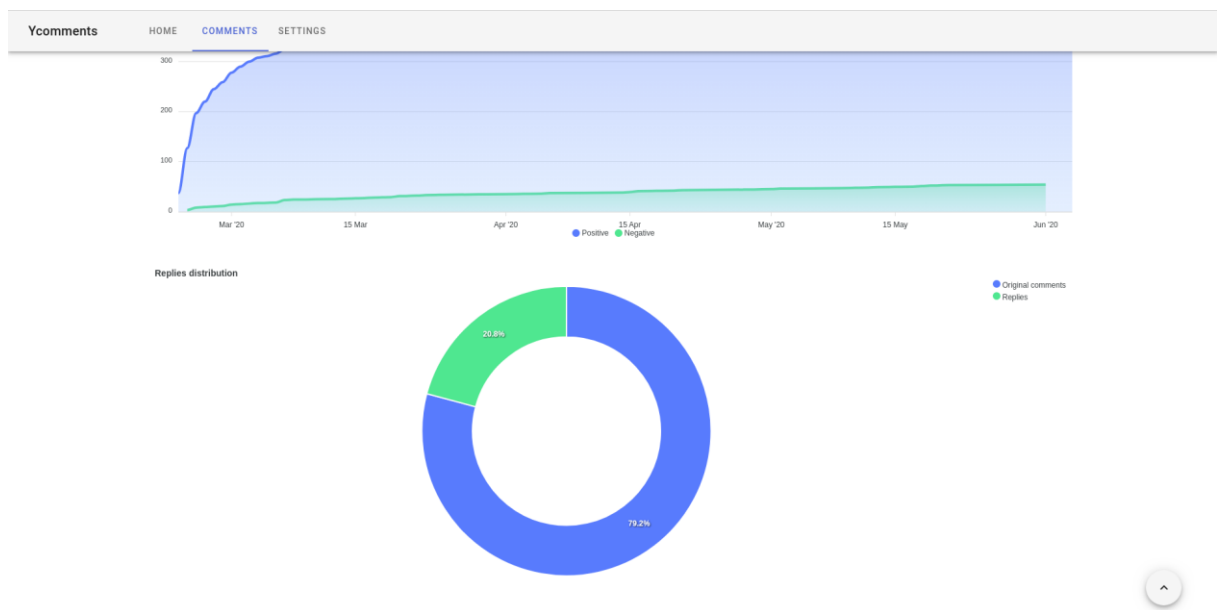


Рис. 15. Графік «Replies distribution»

6. Опис сторінки налаштувань

При натисканні на вкладку «Settings», що знаходиться на навігаційній панелі веб-сервісу, користувач потрапляє на сторінку налаштувань (рис. 16). На даній сторінці можна змінити світлу тему на темну (рис. 17) та навпаки, задати YouTube API v3.0 токен, ввівши його у відповідну графу та натиснувши кнопку «SET TOKEN» або вийти із аккаунту.

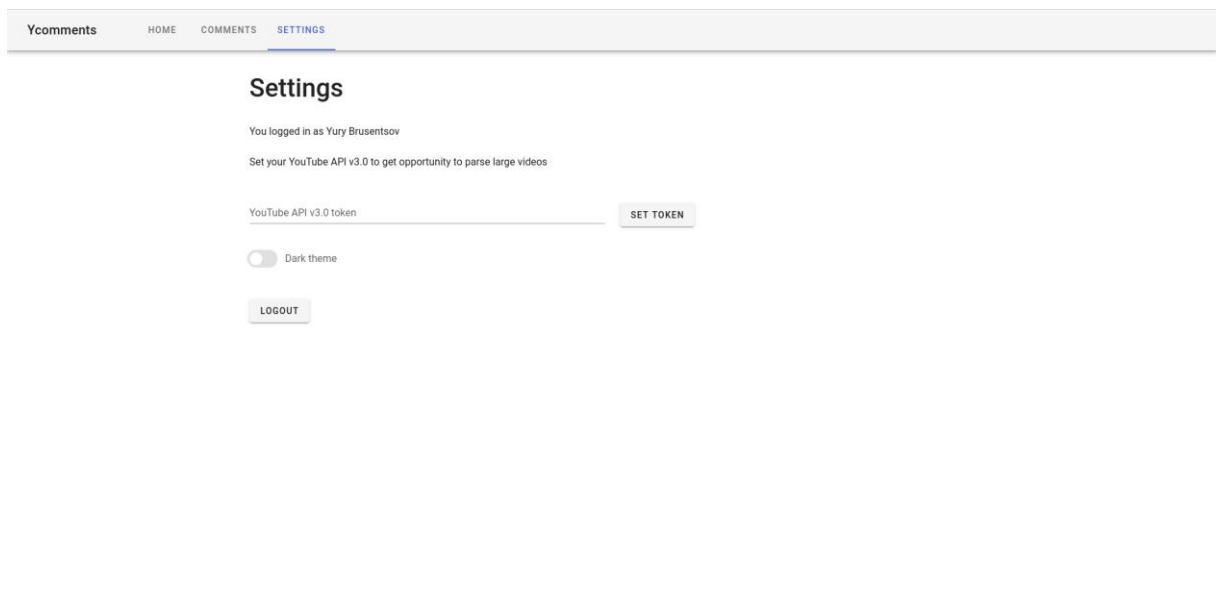


Рис. 16. Сторінка налаштувань

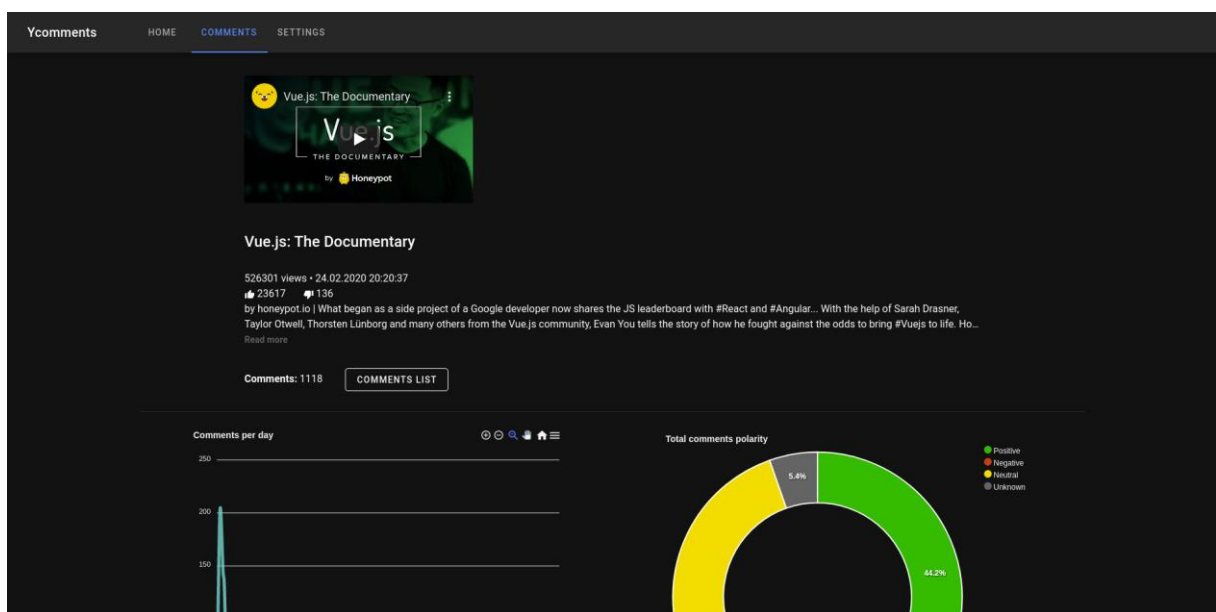
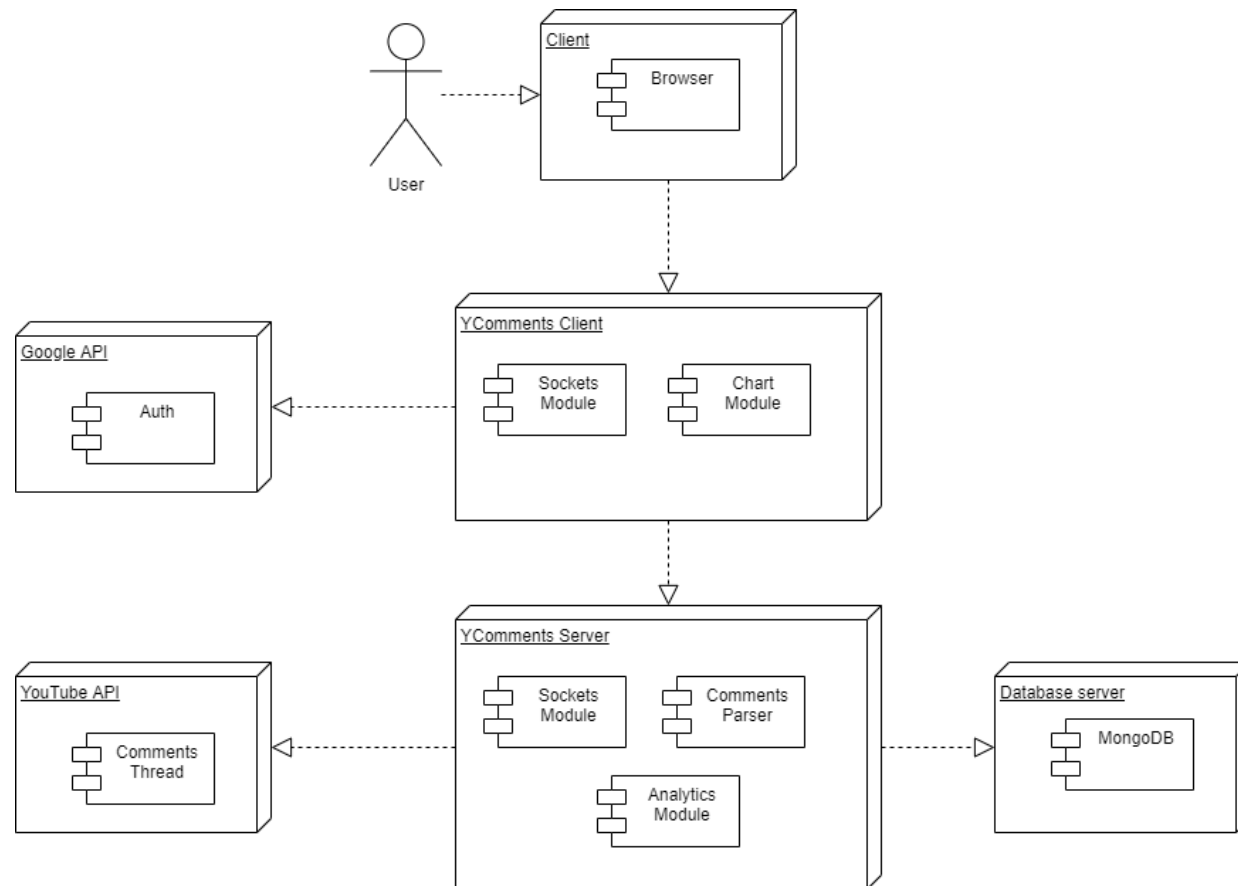


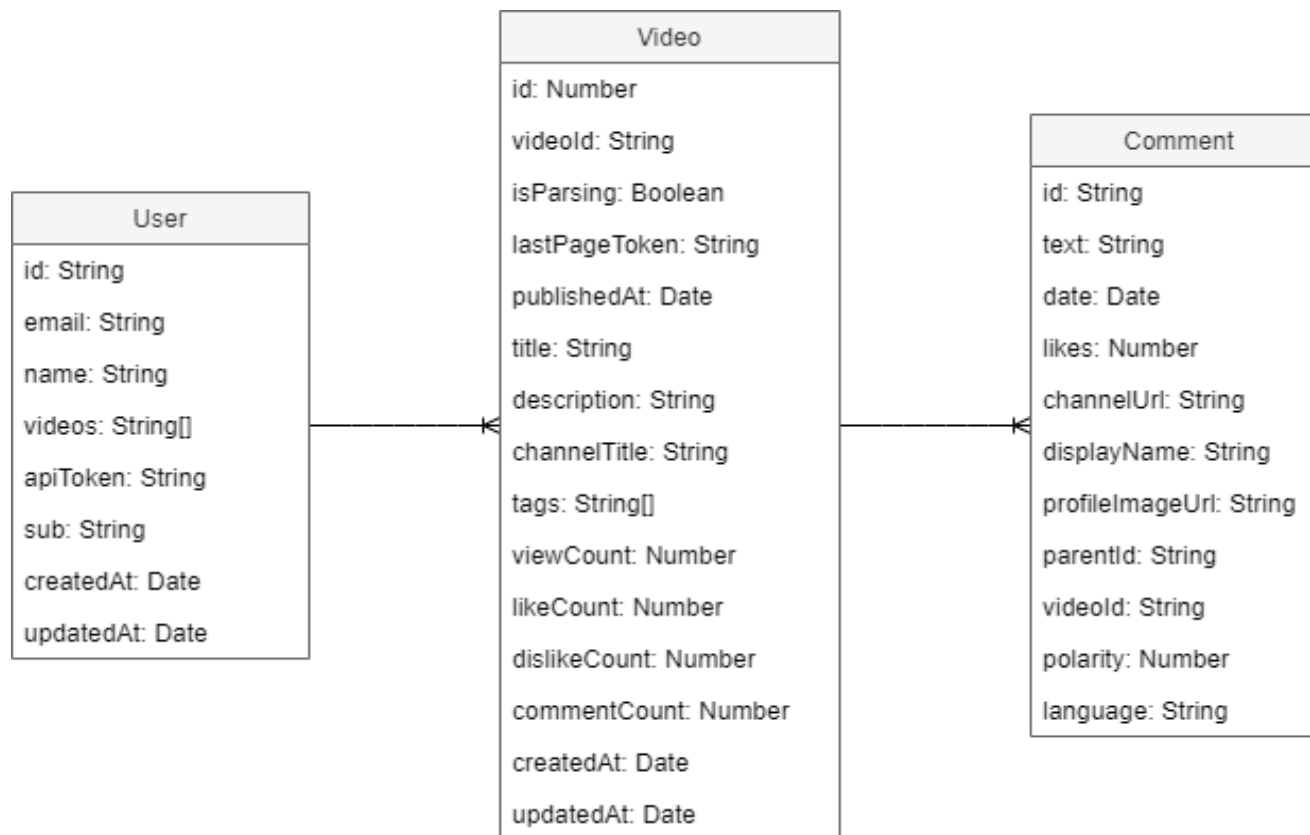
Рис. 17. Сторінка аналізу коментарів із застосуванням темної теми

Додаток 1
Копії графічних матеріалів



ДП.045440-06-99.

Веб-сервіс для аналізу коментарів
на YouTube. Архітектура системи.
Діаграма компонентів веб-сервісу.
ER-діаграма

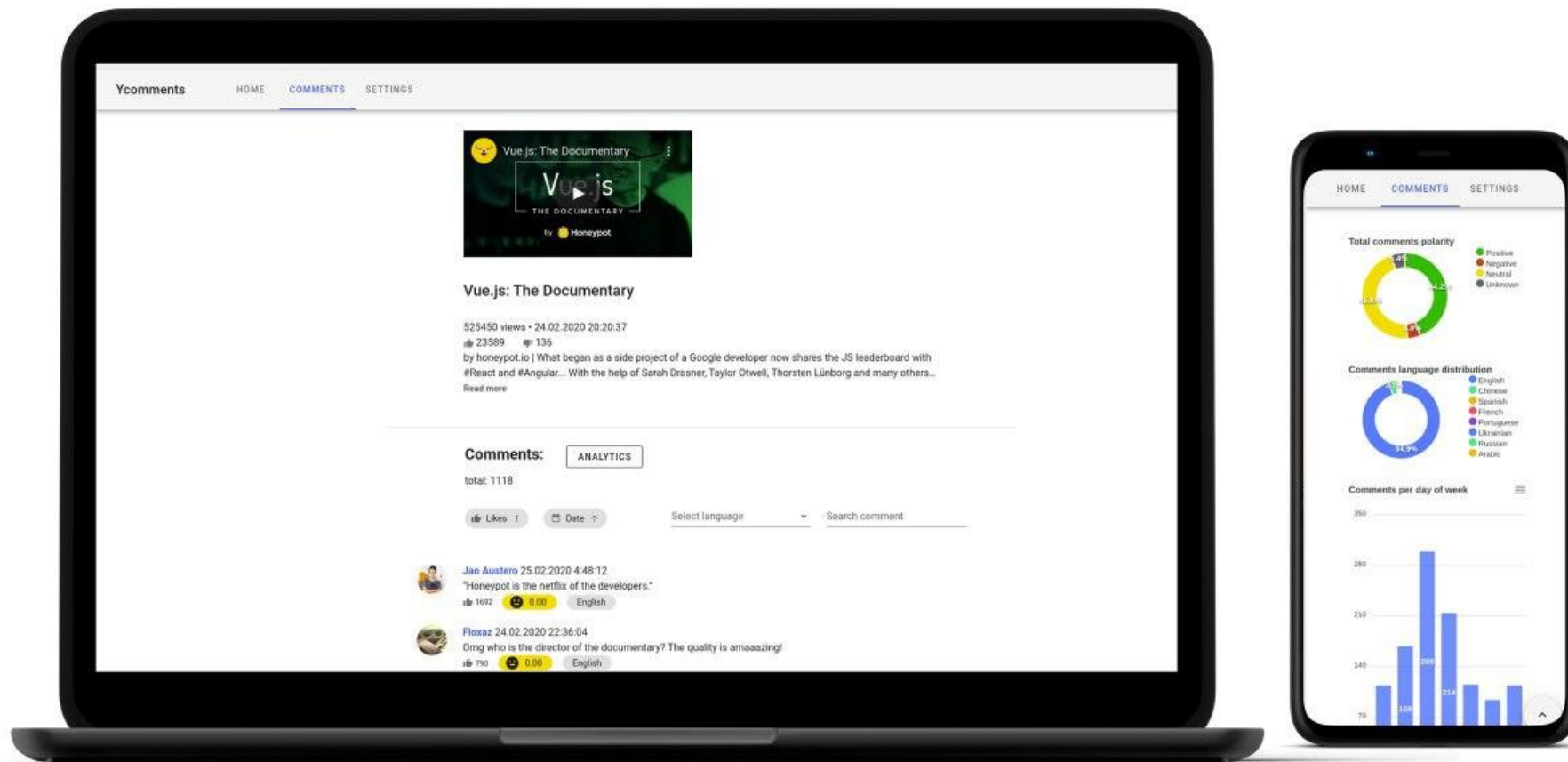


ДП.045440-07-99.

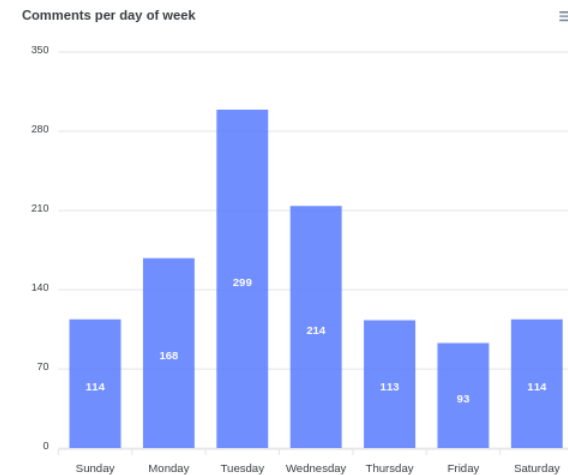
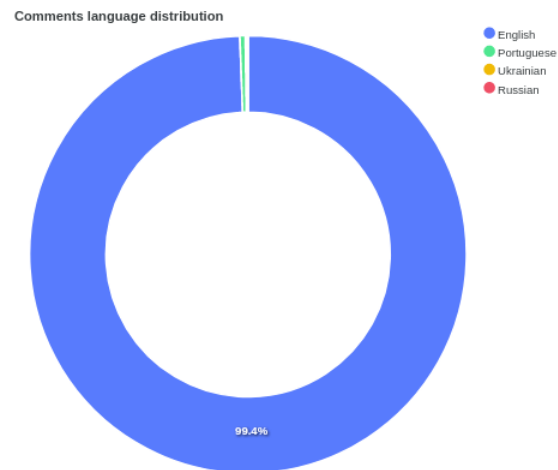
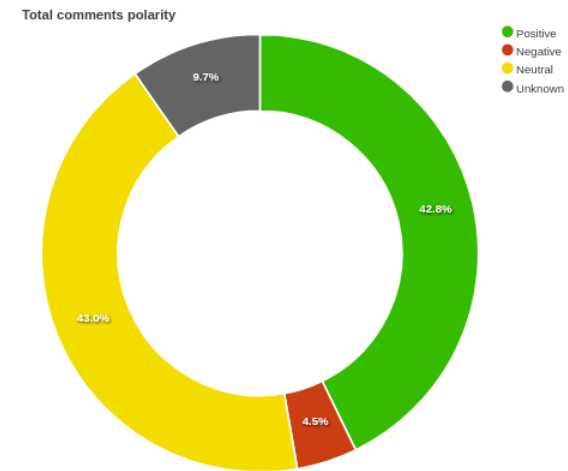
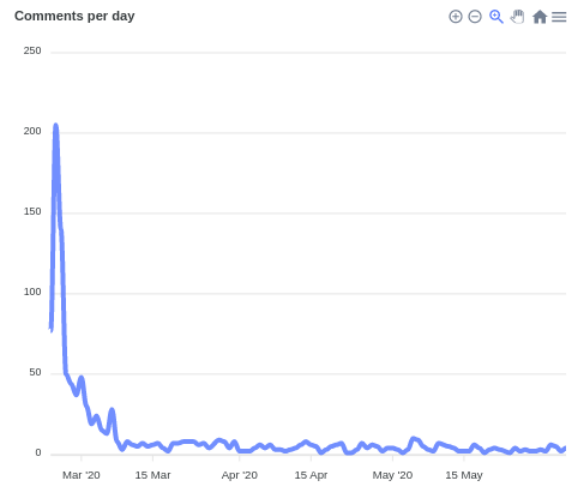
Веб-сервіс для аналізу коментарів
на YouTube. Архітектура системи

Схема бази даних веб-сервісу.

ER-діаграма



Брусенцов Юрій Олексійович
гр. КП-61



Брусенцов Юрій Олексійович
гр. КП-61

Додаток 2
Текст програми

2.1. Функція для збирання коментарів із YouTube Data API v3

```
const parseCommentsPage = ({
  videoId, io = null, key = null, pageToken = null,
}) => {
  const PARAMS = {
    key,
    part: 'snippet,replies',
    videoId,
    maxResults: 100,
    pageToken,
  };

  return request({
    url: COMMENT_THREADS_URL,
    qs: PARAMS,
    json: true,
  })
    .then(async (results) => {
      const commentsArray = [];
      const repliesPromises = [];
      for (let i = 0; i < results.items.length; i += 1) {
        const apiComment = results.items[i];
        const commentObj = getCommentFromApiItem(results.items[i]);
        commentsArray.push(commentObj);
        if ('replies' in apiComment) {
          const repliesTotal = apiComment.snippet.totalReplyCount;
          if (repliesTotal <= 5) { // we can get comments without
            additional requests
              for (let rIndex = 0; rIndex <
                apiComment.replies.comments.length; rIndex += 1) {
                const reply =
                  getCommentFromApiItem(apiComment.replies.comments[rIndex]);
                reply.videoId = videoId;
                commentsArray.push(reply);
              }
            } else { // get reply comments
              repliesPromises.push(parseReplies({ videoId, parentId:
                apiComment.id, key }));
            }
          }
        }
        const analyzedComments = await
          analyzeComments(JSON.stringify(commentsArray));
        await Promise.all(repliesPromises);
        return database.createComments(analyzedComments, videoId);
      })
    .then(async () => {
      await database.setVideoStatus({ videoId, pageToken, isParsing:
        true });
      if (io) { // send intermediate info
```

```

        await sendParsingStatus(io, videoId, true);
    }
    if ('nextPageToken' in results) {
        const { nextPageToken } = results;
        return parseCommentsPage({
            videoId, io, key, pageToken: nextPageToken,
        });
    }
    return pageToken; // parsing is over
})
.catch((error) => Promise.reject(error));
};

```

2.2. Функція для збирання відповідей із YouTube Data API v3

```

const parseReplies = ({
    videoId, parentId, key, pageToken,
}) => {
    const replyParams = {
        fields: ['nextPageToken', 'items'].join(','),
        key,
        part: 'snippet',
        maxResults: 100,
        parentId,
        pageToken,
    };
    return request({
        url: COMMENTS_URL,
        qs: replyParams,
        json: true,
    })
    .then(async (replyResults) => {
        const repliesArray = [];
        for (let i = 0; i < replyResults.items.length; i += 1) {
            const reply = getCommentFromApiItem(replyResults.items[i]);
            reply.videoId = videoId;
            repliesArray.push(reply);
        }
        const analyzedReplies = await
        analyzeComments(JSON.stringify(repliesArray));
        return database.createComments(analyzedReplies, videoId);
    })
    .then(() => {
        if ('nextPageToken' in replyResults) {
            const { nextPageToken } = replyResults;
            return parseReplies({ videoId, key, nextPageToken });
        }
        return null; // parsing replies is over
    })
    .catch((error) => Promise.reject(error));
};

```

2.3. Модуль аналізу тональності та мови написання коментарів

```
# coding=utf-8
import sys
import json
from polyglot.text import Text
class TextOverride(Text):
    def polarity(self):
        """Return the polarity score as a float within the range [-1.0,
1.0]"""
        scores = [w.polarity for w in self.words if w.polarity != 0]
        if len(scores) == 0:
            return 0.0
        return sum(scores) / float(len(scores))

comments = json.loads(sys.argv[1])
for comment in comments:
    try:
        text = TextOverride(comment['text'])
        polarity = text.polarity()
        language = text.language.name
        comment["polarity"] = polarity
        comment["language"] = language
    except Exception as error:
        # don't modify comment object, will return original object
        ...

print(json.dumps(comments))
```

2.4. Функція для використання модулю аналізу на платформі Node.js

```
const analyzeComments = (comment) => new Promise((resolve, reject) =>
{
    const options = {
        mode: 'text',
        scriptPath: __dirname,
        args: [comment],
    };
    PythonShell.run('sentiment-analysis.py', options, (error, result) =>
    {
        if (error) {
            reject(error);
        }
        if (result && result.length > 0) {
            resolve(JSON.parse(result[0])); // return modified comment
        } else {
            reject(new Error('Unexpected error'));
        }
    });
});
```

```
});
```

2.5. Функція для отримання середньої кількості коментарів за день тижня

```
const getCommentsCountByDayOfWeek = (videoId) => Comment.aggregate([
  {
    $match: {
      videoId,
      date: {
        $exists: true,
      },
    },
  },
  {
    $group: {
      _id: { $dayOfWeek: { date: '$date' } },
      comments: { $sum: 1 },
    },
  },
  { $sort: { _id: 1 } },
  {
    $project: {
      _id: 0,
      comments: 1,
      dayOfWeek: '$_id',
    },
  },
])
.then((data) => {
  if (!data || data.length === 0) {
    return Promise.reject(new Error('Wrong videoId or video does not have comments'));
  }
  return Promise.resolve(data);
})
.catch((error) => Promise.reject(error));
```

2.6. Функція для отримання розподілу кількості коментарів за тональністю їх тексту

```
const getCommentsPolarityDistribution = async (videoId) => {
  const total = await Comment.countDocuments({ videoId });
  if (!total) {
    return Promise.reject(new Error('Wrong videoId or video does not have comments'));
  }
}
```

```

    const positive = await Comment.countDocuments({ videoId, polarity: {
$gt: 0 } });
    const negative = await Comment.countDocuments({ videoId, polarity: {
$lt: 0 } });
    const neutral = await Comment.countDocuments({ videoId, polarity: 0
});
    const unknown = total - positive - negative - neutral;
    return {
      videoId,
      total,
      positive,
      negative,
      neutral,
      unknown,
    };
  };
};

```

2.7. Функція для отримання кількості коментарів за добу

```

const getCommentsCountPerDay = (videoId) => Comment.aggregate([
  {
    $match: {
      videoId,
      date: {
        $exists: true,
      },
    },
  },
  {
    $group: {
      _id: { $dateToString: { format: '%Y-%m-%d', date: '$date' } },
      comments: { $sum: 1 },
    },
  },
  {
    $sort: {
      _id: 1,
    },
  },
  {
    $project: {
      _id: 0,
      date: { $dateFromString: { dateString: '$_id' } },
      comments: 1,
    },
  },
])
  .then((data) => {
    if (!data || data.length === 0) {

```

```
        return Promise.reject(new Error('Wrong videoId or video does not  
have comments'));  
    }  
    return Promise.resolve(data);  
  })  
  .catch((error) => Promise.reject(error));
```


Додаток 3
Копія презентації



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ВЕБ-СЕРВІС ДЛЯ АНАЛІЗУ КОМЕНТАРІВ НА YOUTUBE

Виконав: Брусенцов Юрій Олексійович

Керівник: ст. викладач, к.т.н., Люшенко Леся Анатоліївна

Київ – 2020

ПОСТАНОВКА ЗАДАЧІ

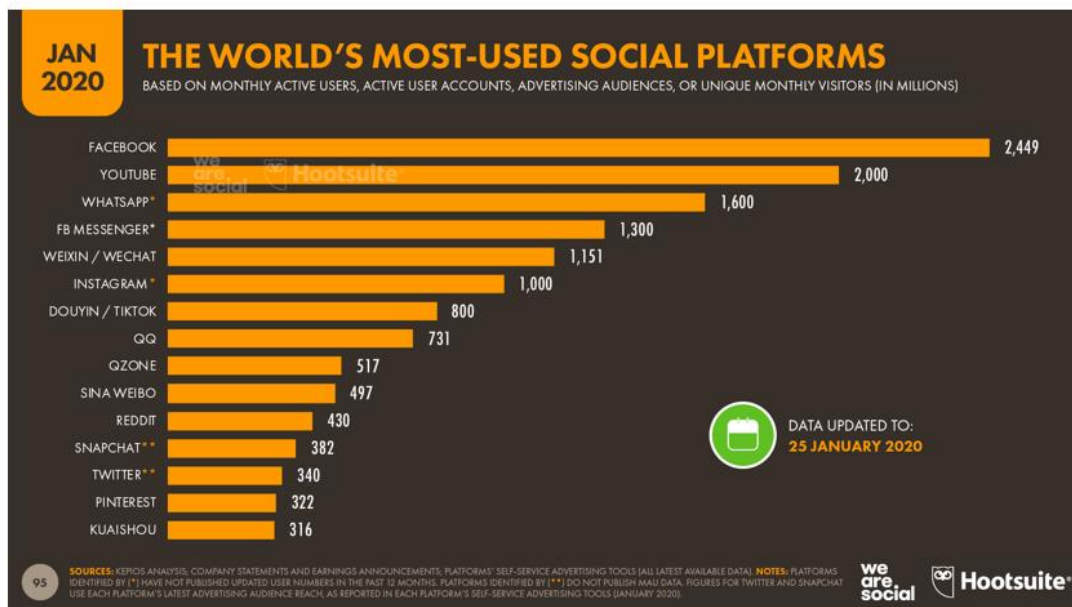


Мета проєкту: створення веб-сервісу, призначеного для спрощення та автоматизації аналізу коментарів із відео на YouTube.

Завдання:

1. Проаналізувати предметну область і визначити її актуальність та проблематику, визначити вимоги до розроблення програмного додатку.
2. Обрати засоби для реалізації ПЗ.
3. Розробити програмний додаток згідно визначених вимог.
4. Протестувати розроблений програмний додаток.

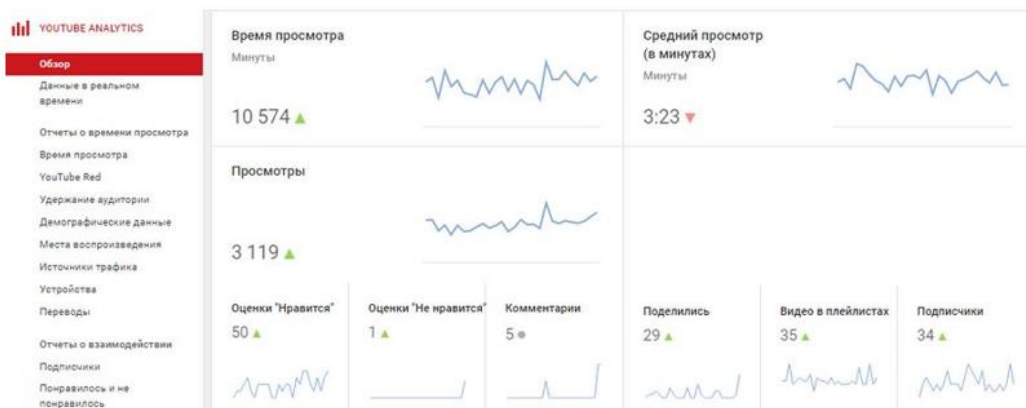
АКТУАЛЬНІСТЬ



3

АКТУАЛЬНІСТЬ

На сьогоднішній день не існує інструмента для повноцінного аналізу коментарів у сервісі YouTube, проте коментарі – це великий та важливий пласт даних, який зберігає у собі інформацію про самих залучених глядачів.



4

АКТУАЛЬНІСТЬ



Типи коментарів:

The screenshot shows a Facebook comment thread. The first comment is from 'Floxx' 3 months ago, asking about the director of a documentary. It has 796 likes and a 'Скрыть 8 ответов' button. The second comment is from 'Josiah McGarvie' 3 months ago, praising the film. The third comment is from 'Evan You' 3 months ago, replying to Josiah. Annotations 1, 2, and 3 are placed next to the first, second, and third comments respectively.

- 1 - Первинний коментар
- 2, 3 - Коментар-відповідь

5

АКТУАЛЬНІСТЬ



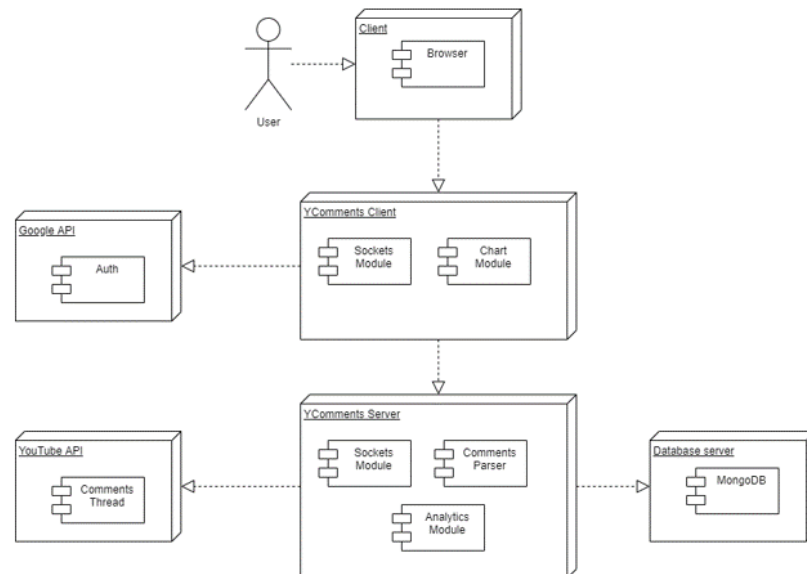
Зміст коментарів:

The screenshot shows the same Facebook comment thread as before. Annotations 1, 2, 3, and 4 are placed next to the first, second, third, and fourth comments respectively. Arrows point from the annotations to the corresponding elements in the comment thread.

- 1. Дата створення коментарю
- 2. Текст коментарю
- 3. Автор коментарю
- 4. Кількість лайків коментарю

6

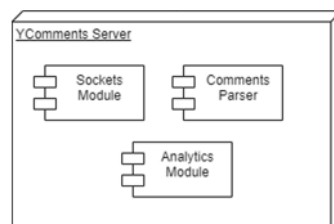
АРХІТЕКТУРА ВЕБ-СЕРВІСУ



7

РОЗРОБЛЕНІ МОДУЛІ

Веб-сервер:

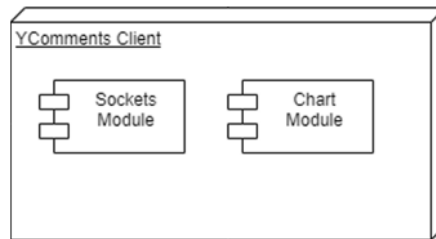


- **CommentsParser** – модуль для збирання коментарів із YouTube Data API v3;
- **SocketsModule** – модуль для взаємодії із веб-клієнтом за допомогою веб-сокетів;
- **AnalyticsModule** – модуль для лінгвістичного аналізу коментарів.

8

РОЗРОБЛЕНІ МОДУЛІ

Веб-клієнт:



- **ChartModule** – модуль для побудови графіків на основі даних взятих із БД;
- **SocketsModule** – модуль для взаємодії із сервером за допомогою веб-сокетів.

9

ЗАСОБИ РОЗРОБЛЕННЯ

Серверна частина



POLYGLOT



10



ЗАСОБИ РОЗРОБЛЕННЯ

Клієнтська частина



11



ЗАСОБИ РОЗРОБЛЕННЯ

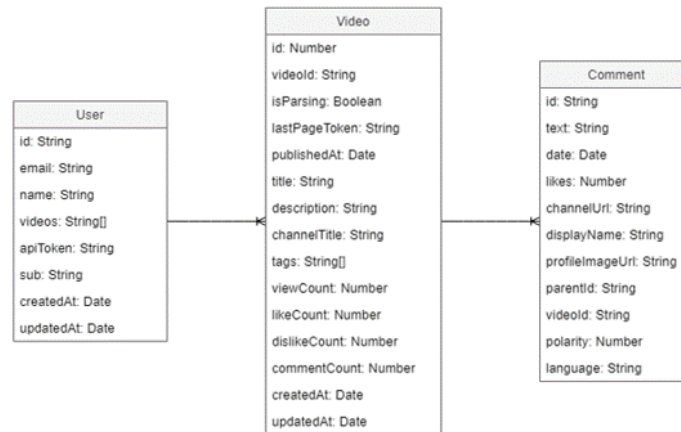
Система керуваннями базами даних



12

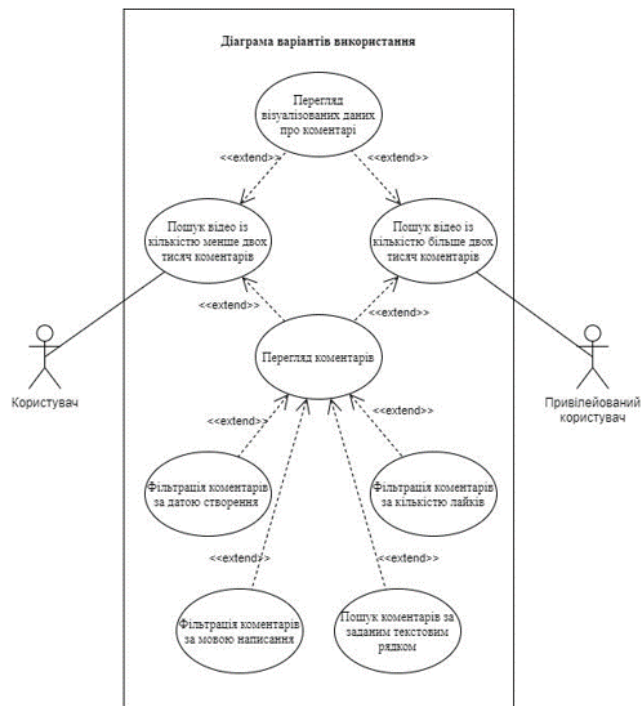
РОЗРОБЛЕНІ МОДУЛІ

База даних:



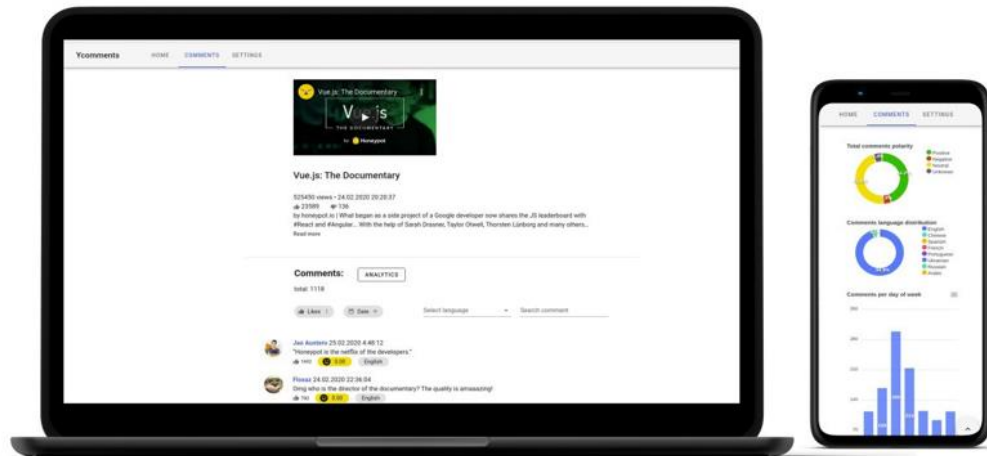
13

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



14

ІНТЕРФЕЙС ВЕБ-СЕРВІСУ

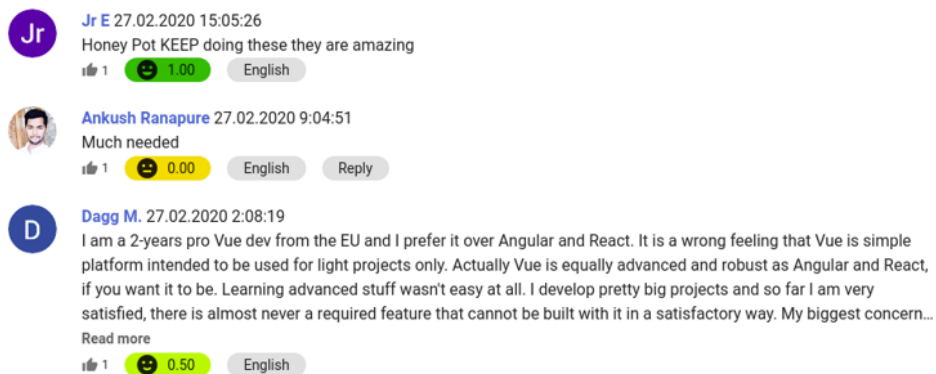


15

РЕЗУЛЬТАТИ РОЗРОБЛЕННЯ ПЗ

У розробленому ПЗ наявні такі функціональні можливості:

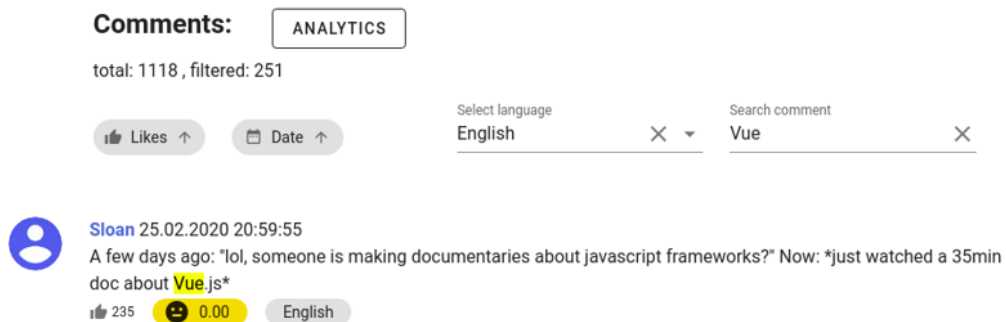
- авторизація користувача за допомогою Google Auth;
- можливість надавати відео для аналізу за допомогою посилання на YouTube;
- відображення коментарів у вигляді списку;



16

РЕЗУЛЬТАТИ РОЗРОБЛЕННЯ ПЗ

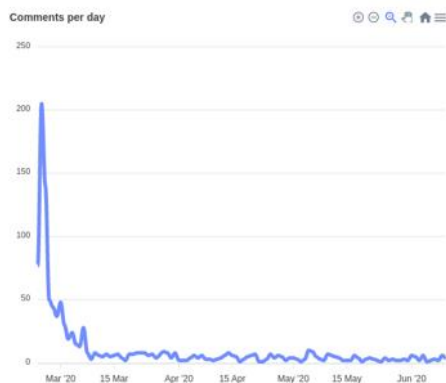
- сортування коментарів за:
 - кількістю лайків;
 - кількістю відповідей на коментар;
 - датою створення коментарю;
- фільтрація коментарів за мовою написання;
- пошук коментарів за входженням текстового рядка;



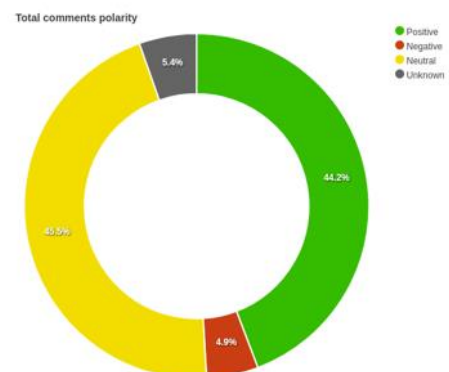
17

РЕЗУЛЬТАТИ РОЗРОБЛЕННЯ ПЗ

Візуалізація статистики коментарів у відео:



кількість коментарів за добу

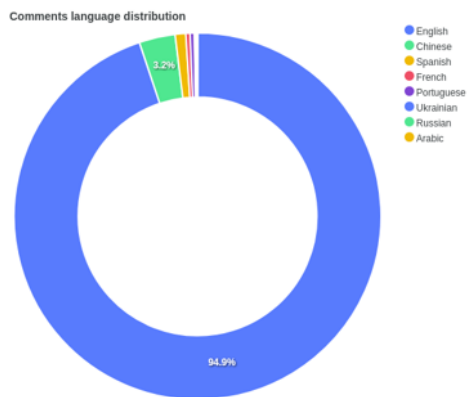


розподіл за емоційним забарвленням

18

РЕЗУЛЬТАТИ РОЗРОБЛЕННЯ ПЗ

Візуалізація статистики коментарів у відео:



розподіл за мовою написання

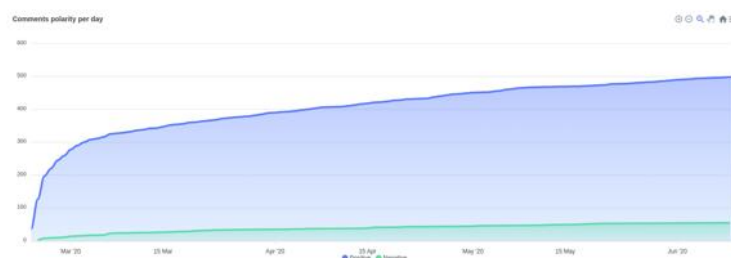


розподіл за днем тижня публікації

19

РЕЗУЛЬТАТИ РОЗРОБЛЕННЯ ПЗ

Візуалізація статистики коментарів у відео:



кількість позитивних та негативних коментарів



відношення первинних коментарів до відповідей

20

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Тестування сервісу проводилося методом «чорного ящика».

Під час тестування було виявлено декілька помилок, наприклад:

- дублювання коментарів-відповідей у списку коментарів;
- неправильне відображення часу створення коментарю;
- невідповідність кількості зібраних коментарів та коментарів під YouTube-відео;

Усі знайдені помилки та недоліки у роботі сервісу було враховано та виправлено.

РЕКОМЕНДАЦІЇ ДЛЯ ПОКРАЩЕННЯ

- дозволити аналізувати відео із будь-якою кількістю коментарів, прибрати роль привілейованого користувача;
- додати можливість аналізу коментарів YouTube-каналу, замість аналізу окремого відео;
- вдосконалити систему пошуку коментарів;
- додати можливість аналізу коментарів прямих трансляцій;
- додати більшу кількість графіків;
- додати систему оповіщень про нові коментарі;
- можливість ставити лайк коментарю безпосередньо із додатку;
- додати можливість аналізувати інші сервіси та соціальні мережі.



УНІКАЛЬНІСТЬ

Назва документу: Брусенцов_КП-61

ID файлу: 1003867305 Кількість сторінок: 37 Кількість слів: 5679 Кількість символів: 44061 Розмір файлу: 208.00 KB

2.18% Схожість

Найбільша схожість: 0.77% з джерело бібліотеки. ID файлу: 1000082006

0.37% Схожість з Інтернет джерелами 2 Page 39

2.18% Текстові збіги по Бібліотеці акаунту 72 Page 39

0% Цитат

Не знайдено жодних цитат

0% Вилучень

Вилучений текст відсутній

Підміна символів

Заміна символів 17

23



ВИСНОВКИ

1. Проаналізовано предметну область, визначено її актуальність та проблематику.
2. Визначено вимоги до розроблення програмного додатку.
3. Розроблено програмне забезпечення для аналізу коментарів на YouTube.
4. Розроблене ПЗ протестовано, визначено шляхи для подальшого вдосконалення сервісу.

24



Дякую за увагу!